

# GemBox Support Center

Knowledgebase > GemBox.Email > How to convert an email to PDF file

---

## How to convert an email to PDF file

Mario - GemBox - 2019-09-10 - 0 Comments - in GemBox.Email

Currently GemBox.Email does not have a support for PDF format. However, our other component, GemBox.Document, supports PDF as both input and output file format (see [Supported File Formats](#) help page).

So, by using both GemBox.Email and GemBox.Document, we can accomplish the conversion of email messages to PDF files.

With GemBox.Email, we can read the MSG, EML, and MHTML files and retrieve their content (subject, body, etc.), while with GemBox.Document we can create a PDF file from that retrieved email content.

The following example demonstrates how to read a [Sample.eml](#) email message, retrieve the message's information and generate a new PDF file from it.

### C# code

```
using System.Linq;
using GemBox.Document;
using GemBox.Email;
using GemBox.Email.Mime;
using HtmlAgilityPack;

class Program
{
    static void Main()
    {
        GemBox.Email.ComponentInfo.SetLicense("FREE-LIMITED-KEY");
        GemBox.Document.ComponentInfo.SetLicense("FREE-LIMITED-KEY");

        var inputEmailPath = "Sample.eml";
        var outputPdfPath = "Sample.pdf";

        // Load email file.
        MailMessage message = MailMessage.Load(inputEmailPath);

        // Create new document.
        DocumentModel document = new DocumentModel();
        document.DefaultParagraphFormat.SpaceAfter = 0;
    }
}
```

```

        // Load email's headers.
        var headers = string.Concat(message.MimeEntity.Headers
            .Select(header => $"{header.Name.PadLeft(16)}:
{header.Body}\n"));
        document.Content.LoadText(headers,
            new CharacterFormat()
            {
                FontName = "Courier New",
                Size = 9
            });

        // Load email's body.
        LoadEmailBody(message, document);

        // Save document as PDF.
        document.Save(outputPdfPath);
    }

    static void LoadEmailBody(MailMessage message, DocumentModel
document)
    {
        if (!string.IsNullOrEmpty(message.BodyHtml))
        {
            // Load HTML content to document.
            var html = ReplaceEmbeddedImages(message.BodyHtml,
message.Attachments);
            document.Content.End.LoadText(html,
LoadOptions.HtmlDefault);
        }
        else
        {
            Attachment attachment =
message.Attachments.SingleOrDefault(
                a => a.MimeEntity.Headers.Any(h => h.Name == "X-Body-
Rtf"));

            if (attachment != null)
            {
                // Load RTF content to document.
                string rtf =
attachment.MimeEntity.Charset.GetString(attachment.MimeEntity.Content

```

```

);
        document.Content.End.LoadText(rtf,
LoadOptions.RtfDefault);
    }

    else
    {
        // Load TXT content to document.
        document.Content.End.LoadText(message.BodyText,
LoadOptions.TxtDefault);
    }
}

}

static string ReplaceEmbeddedImages(string htmlBody,
AttachmentCollection attachments)
{
    // Load email's HTML body using HtmlAgilityPack.
    HtmlDocument htmlDocument = new HtmlDocument() {
OptionEmptyCollection = true };
    htmlDocument.LoadHtml(htmlBody);

    // Traversing through "img" tags and replace "src" content
with embedded image data.
    foreach (HtmlNode imageNode in
htmlDocument.DocumentNode.SelectNodes("//img"))
    {
        HtmlAttribute imageSource = imageNode.Attributes["src"];

        Attachment attachment = attachments.FirstOrDefault(
            a => a.DispositionType ==
ContentDispositionType.Inline &&
            a.ContentId == imageSource.Value.Replace("cid:",
""));

        if (attachment != null)
        {
            var data =
attachment.MimeEntity.Charset.GetString(attachment.MimeEntity.Content
);

            imageSource.Value =

```

```

    $"data:{attachment.MimeEntity.ContentType};base64,{data}";
        }
    }

    return htmlDocument.DocumentNode.OuterHtml;
}
}

```

## VB.NET code

```

Imports System.Linq
Imports GemBox.Document
Imports GemBox.Email
Imports GemBox.Email.Mime
Imports HtmlAgilityPack

```

### Module Program

```

Sub Main()

    GemBox.Email.ComponentInfo.SetLicense("FREE-LIMITED-KEY")
    GemBox.Document.ComponentInfo.SetLicense("FREE-LIMITED-KEY")

    Dim inputEmailPath = "C:\Users\ECHO\Desktop\Sample.eml"
    Dim outputPdfPath = "C:\Users\ECHO\Desktop\AAA_Sample.pdf"

    ' Load email file.
    Dim message As MailMessage = MailMessage.Load(inputEmailPath)

    ' Create new document.
    Dim document As DocumentModel = New DocumentModel()
    document.DefaultParagraphFormat.SpaceAfter = 0

    ' Load email's headers.
    Dim headers = String.Concat(message.MimeEntity.Headers _
        .Select(Function(header) $"{header.Name.PadLeft(16)}: {header.Body}{vbLf}"))
    document.Content.LoadText(headers,
        New CharacterFormat() With
        {
            .FontName = "Courier New",
            .Size = 9
        }
    )
}

```

```

    })

    ' Load email's body.
    LoadEmailBody(message, document)

    ' Save document as PDF.
    document.Save(outputPdfPath)

End Sub

Sub LoadEmailBody(ByVal message As MailMessage, ByVal document As
DocumentModel)

    If Not String.IsNullOrEmpty(message.BodyHtml) Then

        ' Load HTML content to document.
        Dim html = ReplaceEmbeddedImages(message.BodyHtml, messag
e.Attachments)
        document.Content.End.LoadText(html, LoadOptions.HtmlDefau
lt)

    Else

        Dim attachment As Attachment = message.Attachments.Single
OrDefault(
            Function(a) a.MimeEntity.Headers.Any(Function(h) h.Na
me = "X-Body-Rtf"))

        If attachment IsNot Nothing Then
            ' Load RTF content to document.
            Dim rtf As String = attachment.MimeEntity.Charset.Get
String(attachment.MimeEntity.Content)
            document.Content.End.LoadText(rtf, LoadOptions.RtfDef
ault)

        Else
            ' Load TXT content to document.
            document.Content.End.LoadText(message.BodyText, LoadO
ptions.TxtDefault)

        End If

    End If

End Sub

```

End Sub

```
Function ReplaceEmbeddedImages(ByVal htmlBody As String, ByVal attachments As AttachmentCollection) As String
```

```
    ' Load email's HTML body using HtmlAgilityPack.  
    Dim htmlDocument As HtmlDocument = New HtmlDocument() With {.OptionEmptyCollection = True}  
    htmlDocument.LoadHtml(htmlBody)
```

```
    ' Traversing through "img" tags and replace "src" content with embedded image data.
```

```
    For Each imageNode As HtmlNode In htmlDocument.DocumentNode.SelectNodes("//img")
```

```
        Dim imageSource As HtmlAttribute = imageNode.Attributes("src")
```

```
        Dim attachment As Attachment = attachments.FirstOrDefault(  
    (
```

```
        Function(a) a.DispositionType = ContentDispositionType.Inline AndAlso  
            a.ContentId = imageSource.Value.Replace("cid:", ""))
```

```
        If attachment IsNot Nothing Then
```

```
            Dim data = attachment.MimeEntity.Charset.GetString(attachment.MimeEntity.Content)
```

```
            imageSource.Value = $"data:{attachment.MimeEntity.ContentType};base64,{data}"
```

```
        End If
```

```
    Next
```

```
    Return htmlDocument.DocumentNode.OuterHtml
```

```
End Function
```

```
End Module
```

This is the resulting [Sample.pdf](#) file:

