**How to Add Custom Fonts When Converting HTML to PDF in .NET Core**

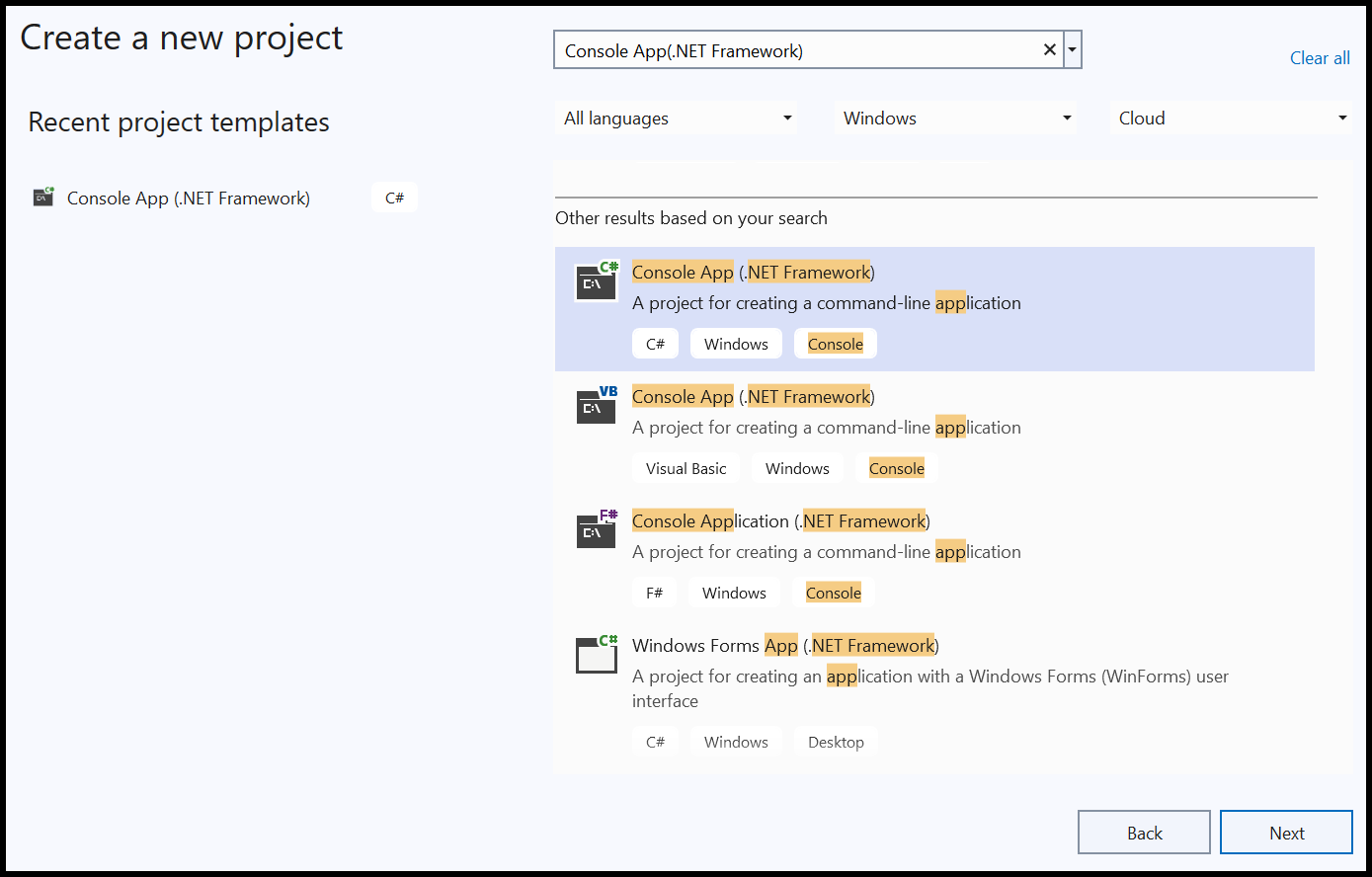
Our Syncfusion® [**HTML-to-PDF converter**](https://www.syncfusion.com/document-processing/pdf-framework/net/html-to-pdf) is a .NET PDF library for converting webpages, SVG MHTML and HTML files to PDF using C#. It uses the popular rendering engine Blink (Google Chrome). It is reliable and accurate. The result preserves all graphics, images, text, fonts, and the layout of the original HTML document or webpage.

You have the flexibility to customize fonts in header and footer elements of your generated PDF. This can be achieved by utilizing the **[PdfTrueTypeFont](https://help.syncfusion.com/cr/document-processing/Syncfusion.Pdf.Graphics.PdfTrueTypeFont.html" \t "_blank)** class from the **[Syncfusion.Pdf.Graphics](https://help.syncfusion.com/cr/document-processing/Syncfusion.Pdf.Graphics.html" \t "_blank)** namespace.

To apply a custom font, you first need to load the desired font from a file and pass it into the PdfTrueTypeFont constructor. The PdfTrueTypeFont constructor accepts parameters like the font file path, font size, and style (e.g., bold, italic). Once the font is initialized, you can assign it to the header or footer elements of the PDF.

**Steps to adding custom font in footer while performing HTML to PDF conversion**

1. Create a new console application project.



1. Install the **[Syncfusion.HtmlToPdfConverter.Winforms](https://www.nuget.org/packages/Syncfusion.HtmlToPdfConverter.WinForms" \t "_blank)** NuGet package as a reference to your console application from [**Nuget.org**](https://www.nuget.org/).

A screenshot of a computer

AI-generated content may be incorrect.

1. Include the following namespaces in the Program.cs file.

**C#**

**using** System.Drawing;

**using** Syncfusion.HtmlConverter;

**using** Syncfusion.Pdf;

**using** Syncfusion.Pdf.Graphics;

1. Use the following code sample in Program.cs to add custom font in footer while performing HTML to PDF conversion  
   **C#**

// Initialize the HTML to PDF converter

HtmlToPdfConverter htmlConverter = **new** HtmlToPdfConverter();

BlinkConverterSettings blinkConverterSettings = **new** BlinkConverterSettings();

// Create font and brush for the footer element

FileStream fontStream = **new** FileStream(@"calibri.ttf", FileMode.Open, FileAccess.Read);

PdfFont font = **new** PdfTrueTypeFont(fontStream, 10);

PdfBrush brush = **new** PdfSolidBrush(Color.Black);

// Create a PDF page template element for the footer with specified bounds

PdfPageTemplateElement footer = **new** PdfPageTemplateElement(**new** RectangleF(0, 0, blinkConverterSettings.PdfPageSize.Width, 50));

// Create a page number field for the footer

PdfPageNumberField pageNumber = **new** PdfPageNumberField(font, PdfBrushes.Black);

// Create a page count field for the footer

PdfPageCountField count = **new** PdfPageCountField(font, PdfBrushes.Black);

// Add page number and page count fields to a composite field for formatted display

PdfCompositeField compositeField = **new** PdfCompositeField(font, PdfBrushes.Black, "Page {0} of {1}", pageNumber, count);

// Draw the composite field in the footer

compositeField.Draw(footer.Graphics, PointF.Empty);

// Assign the footer element to the PDF footer settings of the Blink converter

blinkConverterSettings.PdfFooter = footer;

// Set the Blink viewport size

blinkConverterSettings.ViewPortSize = **new** Size(1024, 0);

// Assign the Blink converter settings to the HTML converter

htmlConverter.ConverterSettings = blinkConverterSettings;

// Convert the specified URL to PDF

PdfDocument document = htmlConverter.Convert("https://www.google.com/");

// Create a file stream to save the generated PDF

**using**(FileStream fileStream = **new** FileStream("Output.pdf", FileMode.Create, FileAccess.ReadWrite))

{

// Save the PDF document to the file stream

document.Save(fileStream);

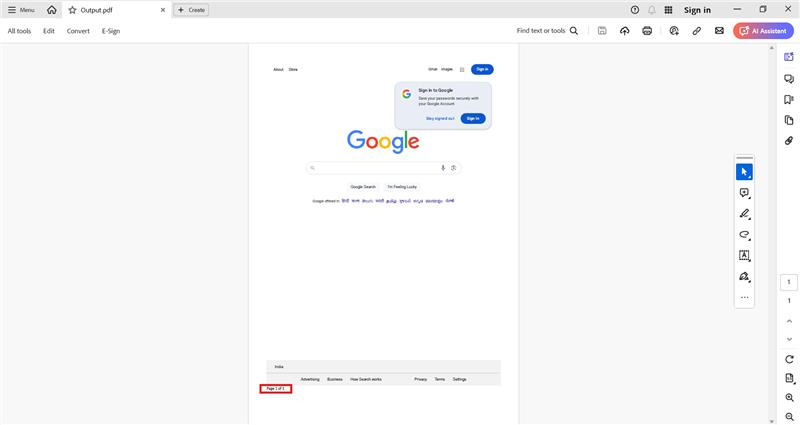
}

// Close the document

document.Close(true);

A complete working sample can be downloaded from [**HTML-to-PDF-Framework-Footer-Custom-Font**](https://www.syncfusion.com/downloads/support/directtrac/general/ze/HTML-to-PDF-Framework-Footer-Custom-Font-1281783883)

By executing the program, a PDF file will be generated with the footer text displayed in the Calibri font, as shown below.



Take a moment to peruse the [**documentation**](https://help.syncfusion.com/document-processing/pdf/conversions/html-to-pdf/net/features#header-and-footer), where you can find converting HTML to PDF document with Header and Footer.

**Conclusion**  
I hope you enjoyed learning about how to add custom fonts when converting HTML to PDF in .NET Core.

You can refer to our [**ASP.NET Core PDF feature tour**](https://www.syncfusion.com/document-processing/pdf-framework/net-core) page to know about its other groundbreaking feature representations and [**documentation**](https://help.syncfusion.com/file-formats/pdf/create-pdf-file-in-asp-net-core), and how to quickly get started for configuration specifications. You can also explore our [**ASP.NET Core PDF example**](https://ej2.syncfusion.com/aspnetcore/PDF/Default#/bootstrap5) to understand how to create and manipulate data.

For current customers, you can check out our components from the [**License and Downloads**](https://www.syncfusion.com/sales/teamlicense) page. If you are new to Syncfusion®, you can try our 30-day [**free trial**](https://www.syncfusion.com/downloads/aspnetcore-js2) to check out our other controls.

If you have any queries or require clarifications, please let us know in the comments section below. You can also contact us through our [**support forums**](https://www.syncfusion.com/forums), [**Direct-Trac**](https://support.syncfusion.com/create), or [**feedback portal**](https://www.syncfusion.com/feedback/aspnet-core?control=pdf). We are always happy to assist you!