**How to embed ZUGFeRD information in ASP.NET Core PDF document**

The Syncfusion Essential® PDF is a feature-rich and high performance [**ASP.NET CorePDF library**](https://www.syncfusion.com/document-processing/pdf-framework/net) used to create, read, and edit PDF documents programmatically without Adobe dependencies. Using this library, you can add ZUGFeRD information in an existing PDF document using C#.

**Steps to add ZUGFeRD information in an existing PDF document programmatically:**

1. Create a new console application project.A screenshot of a computer

   AI-generated content may be incorrect.
2. Install the **[Syncfusion.Pdf.Imaging.Net.Core](https://www.nuget.org/packages/Syncfusion.Pdf.Imaging.Net.Core" \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 **Syncfusion**.Pdf.Parsing;

using **Syncfusion**.Pdf;

using **Syncfusion**.Pdf.Graphics;

using **SkiaSharp**;

using **Syncfusion**.Pdf.Interactive;

1. Use the following code sample in Program.cs to add a ZUGFeRD compliance in an existing PDF document.

**C#**

//Load an existing PDF document

**FileStream** fileStream = **new** **FileStream**(@"../../../Invoice.pdf", **FileMode**.Open, **FileAccess**.Read);

**PdfLoadedDocument** loadedDocument = **new** **PdfLoadedDocument**(fileStream);

//Subscripe the SubstitueFont event for supplying font.

loadedDocument.SubstituteFont += **LoadedDocument**\_SubstituteFont;

//Convert the existing PDF to PDF/A-3B document.

loadedDocument.**ConvertToPDFA**(**Syncfusion**.Pdf.PdfConformanceLevel.Pdf\_A3B);

**MemoryStream** stream = **new** **MemoryStream**();

//Save and close the document

loadedDocument.**Save**(stream);

loadedDocument.**Close**(true);

//Load the converted PDF stream

loadedDocument = **new** **PdfLoadedDocument**(stream);

//Create a new PDF document with A3B

**PdfDocument** document = **new** **PdfDocument**(**PdfConformanceLevel**.Pdf\_A3B);

//Merge the existing document

**PdfDocumentBase**.**Merge**(document, loadedDocument);

//Set the ZUGFeRD conformance level

document.ZugferdConformanceLevel = **ZugferdConformanceLevel**.Basic;

//Set the ZUGFeRD version

document.ZugferdVersion = **ZugferdVersion**.ZugferdVersion2\_0;

//Get stream from the XML file

**FileStream** invoiceStream = **new** **FileStream**(**Path**.**GetFullPath**("../../../ZUGFeRD\_invoice.xml"), **FileMode**.Open, **FileAccess**.Read);

//Create an attachment

**PdfAttachment** attachment = **new** **PdfAttachment**("zugferd-invoice.xml", invoiceStream);

//Add the attachment relationship

attachment.Relationship = **PdfAttachmentRelationship**.Alternative;

//Set modification date

attachment.ModificationDate = **DateTime**.Now;

//Set description and mime type

attachment.Description = "zugferd-invoice";

attachment.MimeType = "text/xml";

//Add the attachment to the document.

document.Attachments.**Add**(attachment);

**MemoryStream** ms = **new** **MemoryStream**();

//Save and close the new PDF document

document.**Save**(ms);

document.**Close**(true);

//Close the loaded document

loadedDocument.**Close**(true);

//Save the PDF file in disk

**File**.**WriteAllBytes**("ZUGFeRD\_PDF.pdf", ms.**ToArray**());

//Add the event to supply the missing fonts

**static** **void** **LoadedDocument**\_SubstituteFont(object sender, **PdfFontEventArgs** args)

{

//get the font name

string fontName = args.FontName.**Split**(',')[0];

//get the font style

**PdfFontStyle** fontStyle = args.FontStyle;

**SKFontStyle** sKFontStyle = **SKFontStyle**.Normal;

**if** (fontStyle != **PdfFontStyle**.Regular)

{

**if** (fontStyle == **PdfFontStyle**.Bold)

{

sKFontStyle = **SKFontStyle**.Bold;

}

**else** **if** (fontStyle == **PdfFontStyle**.Italic)

{

sKFontStyle = **SKFontStyle**.Italic;

}

**else** **if** (fontStyle == (**PdfFontStyle**.Italic | **PdfFontStyle**.Bold))

{

sKFontStyle = **SKFontStyle**.BoldItalic;

}

}

//Get the font stream using Skiasharp

**SKTypeface** typeface = **SKTypeface**.**FromFamilyName**(fontName, sKFontStyle);

**SKStreamAsset** typeFaceStream = typeface.**OpenStream**();

**MemoryStream** memoryStream = null;

**if** (typeFaceStream != null && typeFaceStream.Length > 0)

{

//Create the fontData from the type face stream.

byte[] fontData = **new** byte[typeFaceStream.Length - 1];

typeFaceStream.**Read**(fontData, typeFaceStream.Length);

typeFaceStream.**Dispose**();

//Create the new memory stream from the font data.

memoryStream = **new** **MemoryStream**(fontData);

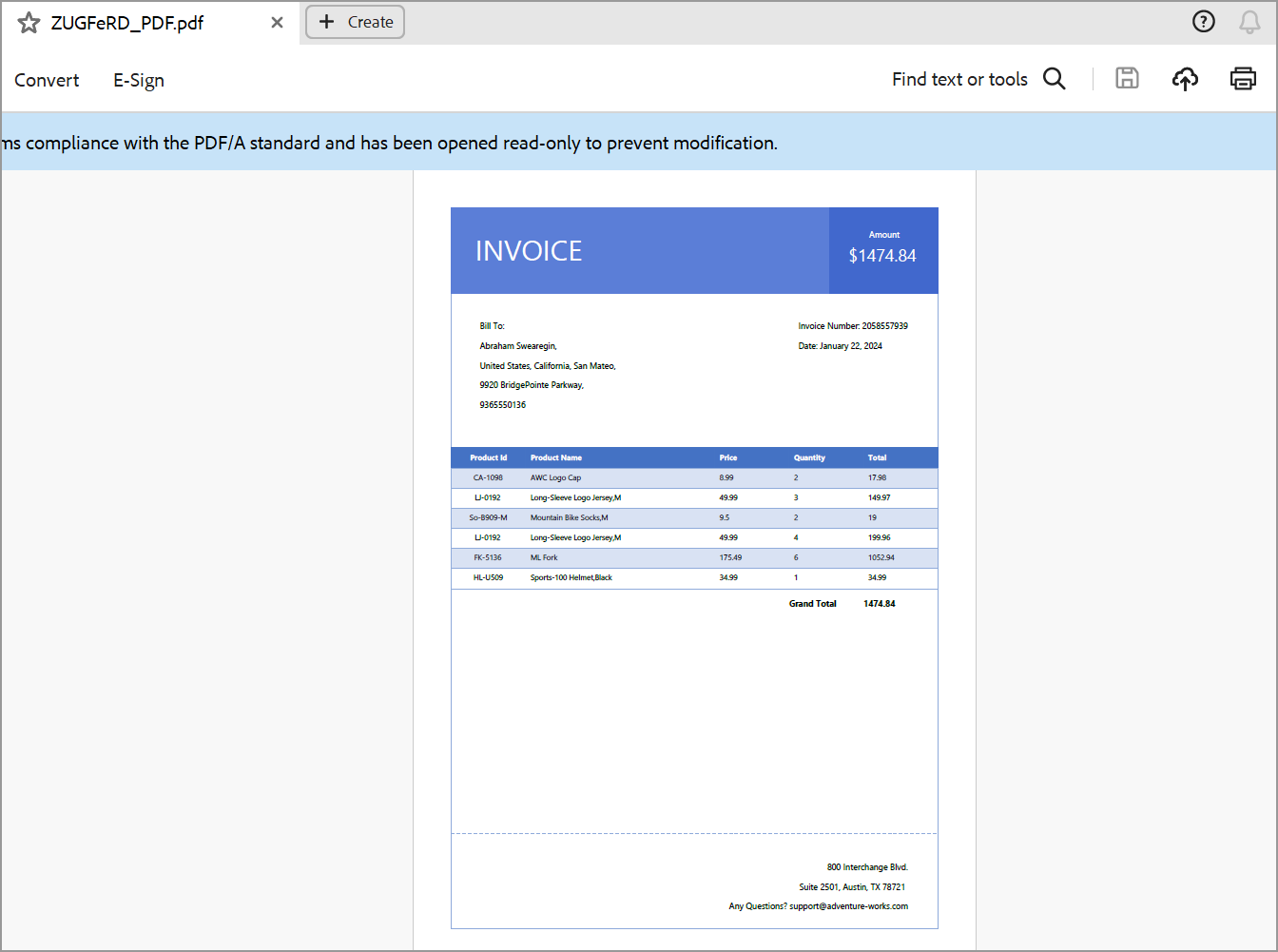
}

//set the font stream to the event args.

args.FontStream = memoryStream;

}

A complete working sample can be downloaded from [**Adding\_ZUGFeRD\_properties.zip**](https://www.syncfusion.com/downloads/support/directtrac/general/ze/Adding_ZUGFeRD_properties1866303799.zip)

By executing the program, you will get the PDF document as follows.

Take a moment to peruse the [**documentation**](https://help.syncfusion.com/file-formats/pdf/working-with-text). You will also find other options like create PDF/A-1b and PDF/X-1a document and features like converting [**Word to PDF**](https://help.syncfusion.com/file-formats/pdf/working-with-document-conversions#converting-word-documents-to-pdf), [**Excel to PDF**](https://help.syncfusion.com/file-formats/pdf/working-with-document-conversions#converting-excel-documents-to-pdf), and [**HTML to PDF**](https://help.syncfusion.com/file-formats/pdf/converting-html-to-pdf) with code examples.

Refer [**here**](https://www.syncfusion.com/document-processing/pdf-framework/net)to explore the rich set of Syncfusion Essential® PDF features.

**Conclusion**

I hope you enjoyed learning about how to embed ZUGFeRD information in [**ASP.NET**](http://asp.net/) Core PDF document.

You can refer to our [**ASP.NET Core PDF**](https://www.syncfusion.com/document-processing/pdf-framework/net) feature tour page to know about its other groundbreaking feature representations and documentation, and how to quickly get started for configuration specifications. You can also explore our [**ASP.NET Core PDF example**](https://www.syncfusion.com/demos/fileformats/pdf-library) to understand how to create and manipulate data.

For current customers, you can check out our components from the [**License and Downloads**](https://ej2.syncfusion.com/angular/documentation/diagram/getting-started) page. If you are new to Syncfusion®, you can try our 30-day [**free trial**](https://www.syncfusion.com/downloads/fileformats) 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!