**How to Apply One or More Digital Signatures to a PDF in ASP.NET Core**

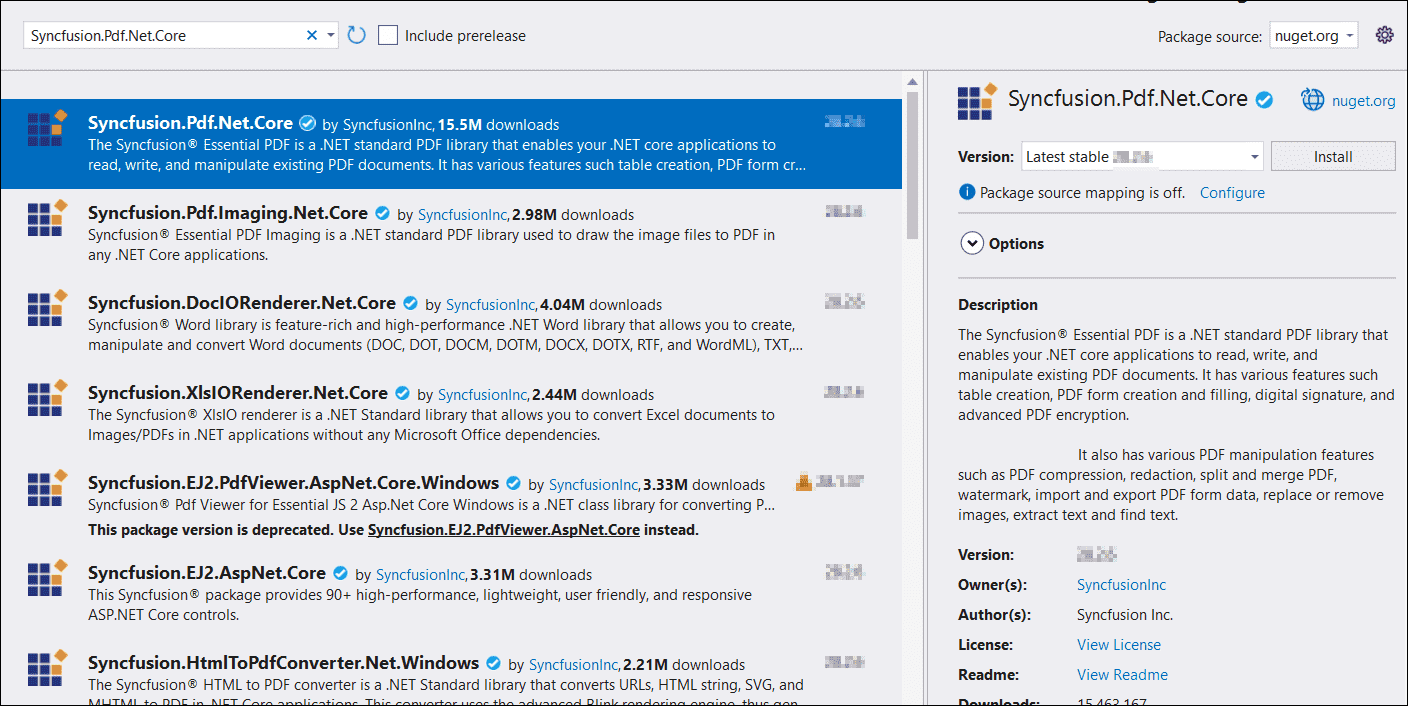
Learn how to use Syncfusion® Essential PDF for [ASP.NET Core](https://www.syncfusion.com/document-processing/pdf-framework/net-core/pdf-library), a powerful library for creating, reading, and editing PDF documents, to apply one or more digital signatures using C# and VB.NET. This guide will take you through the complete process step-by-step.

**Steps to apply one or more digital signatures to a PDF document programmatically**:

1. **Create a New Project**: Start by setting up a new C# ASP.NET Core Web application. Select the Model-View-Controller (MVC) pattern for your project.

A screenshot of a computer

AI-generated content may be incorrect.

2.**Install Necessary Packages**: Add the [Syncfusion.Pdf.Net.Core](https://www.nuget.org/packages/Syncfusion.Pdf.Net.Core/) NuGet package to your .NET Standard application from [NuGet.org](https://www.nuget.org/).  


3**. Configure the Application**: On project creation, a default controller named **HomeController.cs** is added. Include the necessary namespaces

**C#**

|  |
| --- |
| **using** Syncfusion.Pdf;  **using** Syncfusion.Pdf.Parsing;  **using** Syncfusion.Pdf.Security;  **using** Syncfusion.Pdf.Graphics; |

**VB.NET**

|  |
| --- |
| **Imports** Syncfusion.Pdf  **Imports** Syncfusion.Pdf.Parsing  **Imports** Syncfusion.Pdf.Security  **Imports** Syncfusion.Pdf.Graphics |

**4.Implement the PDF Generation**: Find the Index method in **HomeController.cs**. Add a button in **Index.cshtml** to trigger PDF generation.

|  |
| --- |
| <h2>Click the button to generate PDF</h2>  @using (Html.BeginForm("GeneratePDF", "Home", FormMethod.Post))  {  <input type="submit" value="Generate PDF" />  } |

5. **Add Digital Signatures**: Implement the GeneratePDF method in HomeController.cs to apply the digital signatures using the following code.

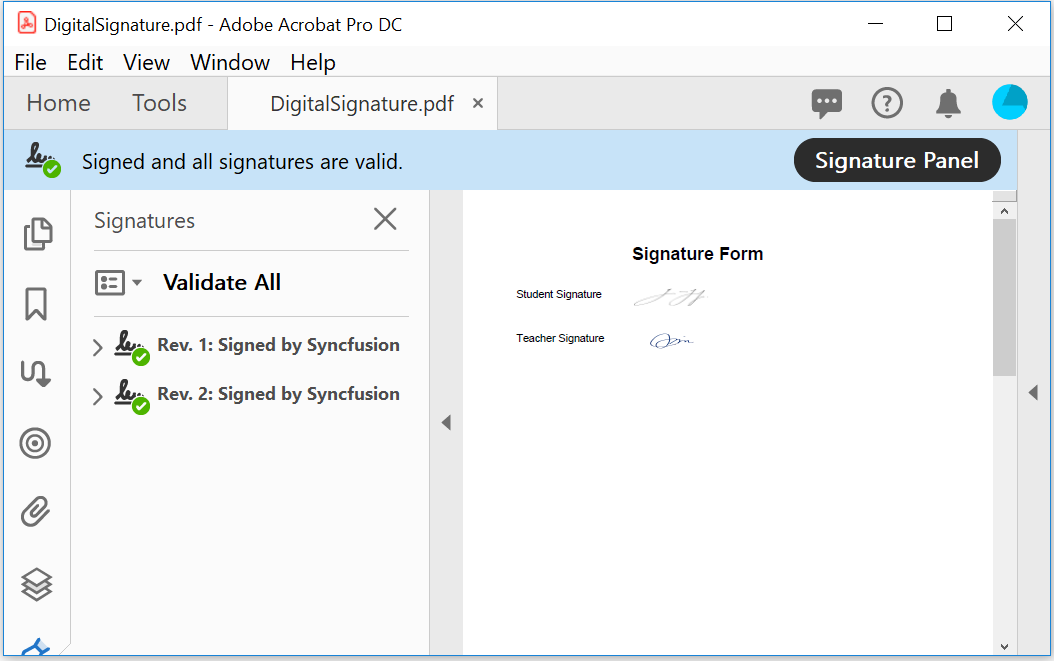
**C#**

|  |
| --- |
| // Load the PDF document  using (FileStream docStream = new FileStream("SignatureFields.pdf", FileMode.Open, FileAccess.Read))  {  PdfLoadedDocument loadedDocument = new PdfLoadedDocument(docStream);  // Get the first page of the document  PdfLoadedPage page = loadedDocument.Pages[0] as PdfLoadedPage;  // Get the first signature field of the PDF document  PdfLoadedSignatureField signatureField1 = loadedDocument.Form.Fields[0] as PdfLoadedSignatureField;  // Create a certificate  using (FileStream certificateStream1 = new FileStream("PDF.pfx", FileMode.Open, FileAccess.Read))  {  PdfCertificate certificate1 = new PdfCertificate(certificateStream1, "syncfusion");  // Apply the signature to the first signature field  signatureField1.Signature = new PdfSignature(loadedDocument, page, certificate1, "Signature", signatureField1);  // Load the student signature image  using (FileStream imageStream = new FileStream("Student Signature.jpg", FileMode.Open, FileAccess.Read))  {  // Draw the student signature image  PdfBitmap signatureImage = new PdfBitmap(imageStream);  signatureField1.Signature.Appearance.Normal.Graphics.DrawImage(signatureImage, 0, 0, 90, 20);  }  // Save the document into a stream  using (MemoryStream stream = new MemoryStream())  {  loadedDocument.Save(stream);  stream.Position = 0;  // Load the signed PDF document  using (PdfLoadedDocument signedDocument = new PdfLoadedDocument(stream))  {  // Get the first page of the signed document  PdfLoadedPage loadedPage = signedDocument.Pages[0] as PdfLoadedPage;  // Get the second signature field of the PDF document  PdfLoadedSignatureField signatureField2 = signedDocument.Form.Fields[1] as PdfLoadedSignatureField;  // Apply the signature to the second signature field  signatureField2.Signature = new PdfSignature(signedDocument, loadedPage, certificate1, "Signature", signatureField2);  // Load the teacher signature image  using (FileStream imageStream1 = new FileStream("Teacher Signature.png", FileMode.Open, FileAccess.Read))  {  // Draw the teacher signature image  PdfBitmap signatureImage1 = new PdfBitmap(imageStream1);  signatureField2.Signature.Appearance.Normal.Graphics.DrawImage(signatureImage1, 0, 0, 90, 20);  }  // Save the final signed PDF to a FileStream  using (FileStream finalFileStream = new FileStream("Output.pdf", FileMode.Create, FileAccess.Write))  {  signedDocument.Save(finalFileStream);  }    }  }  }  } |

**VB.NET**

|  |
| --- |
| ' Load the PDF document  Using docStream As New FileStream("SignatureFields.pdf", FileMode.Open, FileAccess.Read)  Dim loadedDocument As New PdfLoadedDocument(docStream)  ' Get the first page of the document  Dim page As PdfLoadedPage = TryCast(loadedDocument.Pages(0), PdfLoadedPage)  ' Get the first signature field of the PDF document  Dim signatureField1 As PdfLoadedSignatureField = TryCast(loadedDocument.Form.Fields(0), PdfLoadedSignatureField)  ' Create a certificate  Using certificateStream1 As New FileStream("PDF.pfx", FileMode.Open, FileAccess.Read)  Dim certificate1 As New PdfCertificate(certificateStream1, "syncfusion")  ' Apply the signature to the first signature field  signatureField1.Signature = New PdfSignature(loadedDocument, page, certificate1, "Signature", signatureField1)  ' Load the student signature image  Using imageStream As New FileStream("Student Signature.jpg", FileMode.Open, FileAccess.Read)  ' Draw the student signature image  Dim signatureImage As New PdfBitmap(imageStream)  signatureField1.Signature.Appearance.Normal.Graphics.DrawImage(signatureImage, 0, 0, 90, 20)  End Using  ' Save the document into a stream  Using stream As New MemoryStream()  loadedDocument.Save(stream)  stream.Position = 0  ' Load the signed PDF document  Using signedDocument As New PdfLoadedDocument(stream)  ' Get the first page of the signed document  Dim loadedPage As PdfLoadedPage = TryCast(signedDocument.Pages(0), PdfLoadedPage)  ' Get the second signature field of the PDF document  Dim signatureField2 As PdfLoadedSignatureField = TryCast(signedDocument.Form.Fields(1), PdfLoadedSignatureField)  ' Apply the signature to the second signature field  signatureField2.Signature = New PdfSignature(signedDocument, loadedPage, certificate1, "Signature", signatureField2)  ' Load the teacher signature image  Using imageStream1 As New FileStream("Teacher Signature.png", FileMode.Open, FileAccess.Read)  ' Draw the teacher signature image  Dim signatureImage1 As New PdfBitmap(imageStream1)  signatureField2.Signature.Appearance.Normal.Graphics.DrawImage(signatureImage1, 0, 0, 90, 20)  End Using  ' Save the final signed PDF to a FileStream  Using finalFileStream As New FileStream("Output.pdf", FileMode.Create, FileAccess.Write)  signedDocument.Save(finalFileStream)  End Using  End Using  End Using  End Using  End Using |

A complete working sample is available for download:[DigitalSignatureSample.zip](https://www.syncfusion.com/downloads/support/directtrac/general/ze/DigitalSignatureSample-2027118639).

By executing the program, the output PDF document will be generated as shown below:

Note:

Starting with v16.2.0.x, include a valid Syncfusion® license key in your projects. Refer to [this link](https://help.syncfusion.com/common/essential-studio/licensing/overview) for more details. 

**Conclusion**

I hope you enjoyed learning about how to apply one or more digital signatures to a PDF in ASP.NET Core.

You can refer to our [**ASP.NET Core PDF**](https://www.syncfusion.com/document-processing/pdf-framework/net-core) feature tour page to know about its other groundbreaking feature representations and [**documentation**](https://help.syncfusion.com/aspnet-core/pdf/getting-started), 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 in the .NET PDF.

For current customers, you can check out our Document processing libraries from the [**License and Downloads**](https://www.syncfusion.com/account/downloads) 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 ASP.NET Core PDF and other .NET Core 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/169347/response-exported-pdf), [**Direct-Trac**](https://support.syncfusion.com/create), or [**feedback portal**](https://www.syncfusion.com/feedback). We are always happy to assist you!