**Adjust Page Orientation in PDFs Programmatically Using Syncfusion .NET Library**

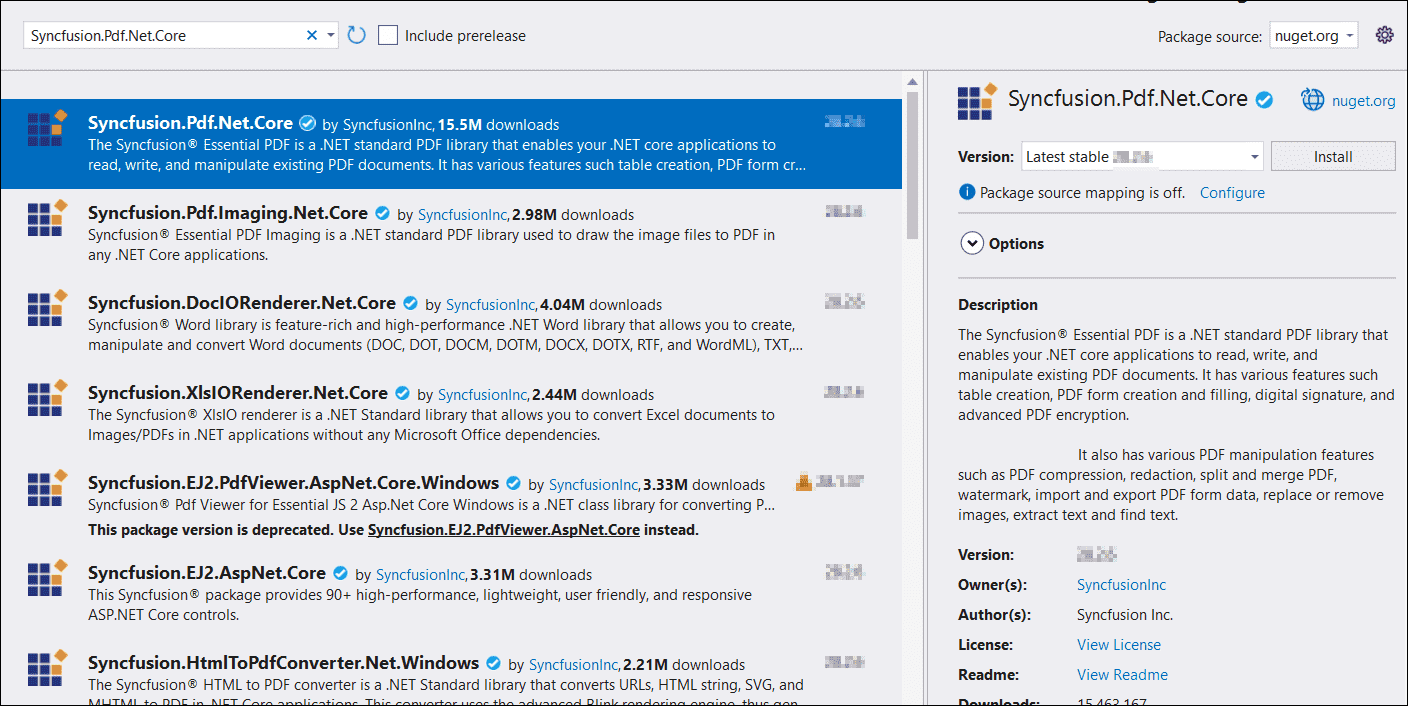
The Syncfusion Essential® PDF is a [**.NET PDF library**](https://www.syncfusion.com/document-processing/pdf-framework/net) used to create, read, and edit PDF documents. This guide demonstrates how to change the orientation of a specific page within a PDF document using C# and VB.NET.

**Steps to Change Page Orientation in a PDF Document**

1. **Set Up Your Project**: Create a new console application project in your development environment.A screenshot of a computer

AI-generated content may be incorrect.

2. **Install Syncfusion Package**: Add the [Syncfusion.Pdf.Net.Core](https://www.nuget.org/packages/Syncfusion.Pdf.Net.Core/) package from NuGet to your project.



3. **Include Required Namespaces**: Add these namespaces to your **Program.cs**.

**C#**

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

[**VB.NET**](http://vb.net/)

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

4. **Code Implementation for Orienting Pages**: Use the following code to adjust the orientation of a specific page in an existing PDF document:

**C#**

|  |
| --- |
| // Load a PDF document.  FileStream docStream = new FileStream("../../../Data/Barcode.pdf", FileMode.Open, FileAccess.Read);  PdfLoadedDocument loadedDocument = new PdfLoadedDocument(docStream);  PdfDocument document = new PdfDocument();  document.PageSettings.Margins.All = 0;  for (int i = 0; i < loadedDocument.Pages.Count; i++)  {  PdfLoadedPage loadedPage = loadedDocument.Pages[i] as PdfLoadedPage;  PdfTemplate template = loadedPage.CreateTemplate();  PdfSection section = document.Sections.Add();  section.PageSettings.Margins.All = 0;  section.PageSettings.Size = new SizeF(loadedPage.Size.Width, loadedPage.Size.Height);    // Change orientation of the first page to Landscape, others to Portrait  section.PageSettings.Orientation = i == 0 ? PdfPageOrientation.Landscape : PdfPageOrientation.Portrait;    PdfPage page = section.Pages.Add();  PdfGraphics graphics = page.Graphics;  graphics.DrawPdfTemplate(template, PointF.Empty, new SizeF(page.Size.Width, page.Size.Height));  }  using (FileStream outputFileStream = new FileStream(Path.GetFullPath(@"Output.pdf"), FileMode.Create, FileAccess.ReadWrite))  {  document.Save(outputFileStream);  }  document.Close(true); |

[**VB.NET**](http://vb.net/)

|  |
| --- |
| ' Load a PDF document.  Dim docStream As New FileStream("../../../Data/Barcode.pdf", FileMode.Open, FileAccess.Read)  Dim loadedDocument As New PdfLoadedDocument(docStream)  Dim document As New PdfDocument()  document.PageSettings.Margins.All = 0  For i As Integer = 0 To loadedDocument.Pages.Count - 1  Dim loadedPage As PdfLoadedPage = TryCast(loadedDocument.Pages(i), PdfLoadedPage)  Dim template As PdfTemplate = loadedPage.CreateTemplate()  Dim section As PdfSection = document.Sections.Add()  section.PageSettings.Margins.All = 0  section.PageSettings.Size = New SizeF(loadedPage.Size.Width, loadedPage.Size.Height)    ' Change orientation of the first page to Landscape, others to Portrait  section.PageSettings.Orientation = If(i = 0, PdfPageOrientation.Landscape, PdfPageOrientation.Portrait)    Dim page As PdfPage = section.Pages.Add()  Dim graphics As PdfGraphics = page.Graphics  graphics.DrawPdfTemplate(template, PointF.Empty, New SizeF(page.Size.Width, page.Size.Height))  Next  Using outputFileStream As New FileStream(Path.GetFullPath("Output.pdf"), FileMode.Create, FileAccess.ReadWrite)  document.Save(outputFileStream)  End Using  document.Close(True) |

**Note:** Changing the document orientation may cause the page content to stretch or shrink, as it adjusts to the new page size dimensions.

A complete working sample can be downloaded from [**ChangePageOrientation.zip**](https://www.syncfusion.com/downloads/support/directtrac/general/ze/Change_Page_Orientation-1855265509.zip).

By executing the program, the output PDF document will be generated as shown below.A screenshot of a computer

AI-generated content may be incorrect.

Take a moment to explore the documentation on [working with pages](https://help.syncfusion.com/file-formats/pdf/working-with-pages), where you'll find options such as inserting, rotating, importing, and rearranging pages from an existing document, as well as removing pages and splitting a PDF file into individual pages

**Conclusion**  
I hope you enjoyed learning about how to change the orientation in specific page using .NET PDF library.

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.