**Creating Stylish PDF Grids with Colored Text Backgrounds Using .NET Core**

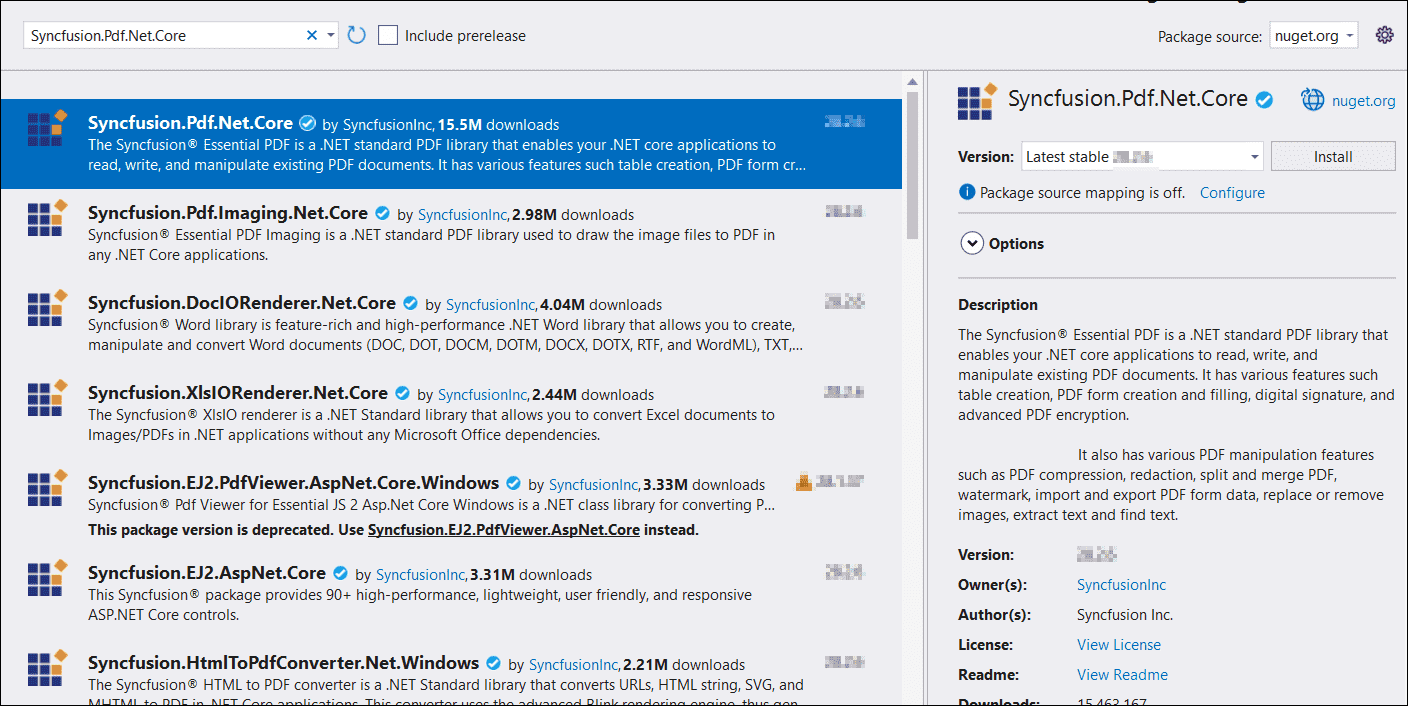
Syncfusion Essential® PDF is a robust [.NET library](https://www.syncfusion.com/document-processing/pdf-framework/net-core) for creating, editing, and managing PDF documents programmatically. This tutorial will guide you through adding text with a colored background in a PDF grid using C#, specifically creating white text on a blue background.

**Steps to Generate Text with a Background in PDF Grid**

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#**

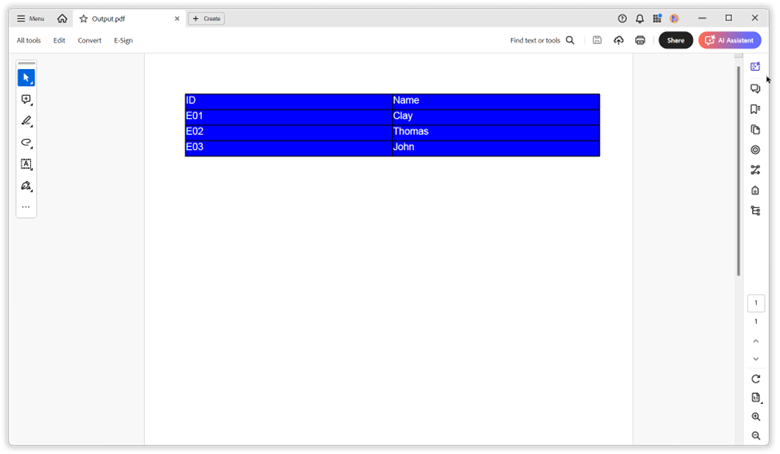
|  |
| --- |
| using **Syncfusion**.Pdf;  using **Syncfusion**.Pdf.Graphics;  using **Syncfusion**.Pdf.Grid; |

4**. Implementation to Add Text with a Background**: Use the following code to style the PDF grid with custom text and background colors

**C#**

|  |
| --- |
| // Create a new PDF document.  PdfDocument document = new PdfDocument();  // Add a page.  PdfPage page = document.Pages.Add();  // Create a PdfGrid.  PdfGrid pdfGrid = new PdfGrid();  // Add values to the list.  List<object> data = new List<object>  {  new { ID = "E01", Name = "Clay" },  new { ID = "E02", Name = "Thomas" },  new { ID = "E03", Name = "John" }  };  // Assign data source.  pdfGrid.DataSource = data;  // Customize cell styles for the grid.  PdfGridCellStyle cellStyle = new PdfGridCellStyle();  // Set background color to blue.  cellStyle.BackgroundBrush = PdfBrushes.Blue;  // Set text color to white.  cellStyle.TextBrush = PdfBrushes.White;  // Set font and size.  cellStyle.Font = new PdfStandardFont(PdfFontFamily.Helvetica, 12f);  // Apply cell style to all cells in the grid.  foreach (PdfGridRow row in pdfGrid.Rows)  {  row.ApplyStyle(cellStyle);  }  // Apply cell style to the header row.  pdfGrid.Headers[0].ApplyStyle(cellStyle);  // Draw the grid to the page of PDF document.  pdfGrid.Draw(page, new Syncfusion.Drawing.PointF(10, 10));  // Create the stream object.  MemoryStream stream = new MemoryStream();  // Save the PDF document to stream.  document.Save(stream);  // Close the document.  document.Close(true);  // Write the PDF document to disk.  File.WriteAllBytes("output.pdf", stream.ToArray()); |

A complete working sample can be downloaded from [**White\_text\_in\_blue\_background.zip**](https://www.syncfusion.com/downloads/support/directtrac/general/ze/White_text_in_blue_background-1815116041.zip)

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

Take a moment to explore the documentation on [working with tables](https://help.syncfusion.com/file-formats/pdf/working-with-tables), where you can find additional features such as grid pagination and advanced grid customization options.

**Conclusion**

I hope you enjoyed learning about how to generate text with a background in PDF grid using C#.

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.