




Print Excel Documents in Just 4 Steps Using C#

 **Mohan Chandran** •  6 min read •  Nov 18, 2024 • **Updated**



Build Faster Apps with Powerful
Excel Processing Libraries

- ✓ Blazing-fast performance
- ✓ 1.7M+ downloads
- ✓ Non-Office dependent

 Syncfusion®

Try it free

No credit card required

Microsoft Excel is a trusted tool for organizing, calculating, and visualizing data. However, the process of converting spreadsheets into tangible documents can sometimes be less straightforward. This is where the Syncfusion [Excel Library](#) becomes invaluable, allowing you to print Excel documents without relying on any Microsoft Excel dependencies.

The [Syncfusion Excel Library](#) (Essential XlsIO) is a robust tool that facilitates the smooth creation, reading, and editing of Excel documents using C#. It supports Excel printing options by converting Excel files to PDF and printing that PDF document. The Excel document can be printed with a specified page setup and printer settings in XlsIO.

Let's see how to print your Excel documents with the help of the Syncfusion Excel Library using C#!



Enjoy a smooth experience with Syncfusion's Excel Library! Get started with a few lines of code and without Microsoft or interop dependencies.

[Explore Now](#)

Excel to PDF

Our XlsIO library supports converting an [entire workbook or a single worksheet into a PDF document](#), using the Syncfusion Excel-to-PDF converter. You can also employ various customization options such as embedding fonts, ignoring empty pages or worksheets, and showing or hiding headers and footers. This process is fast, reliable, and supported in hosting environments such as AWS, Google Cloud App, and Microsoft Azure web services.

This comprehensive library supports the following key features:

- Excel [workbook-to-PDF](#) conversion.
- Excel workbooks with [charts conversion to PDF](#).
- [Font substitution](#) during Excel-to-PDF conversion.
- Multiple Excel-to-PDF [conversion options](#).

Steps to print Excel documents using C#

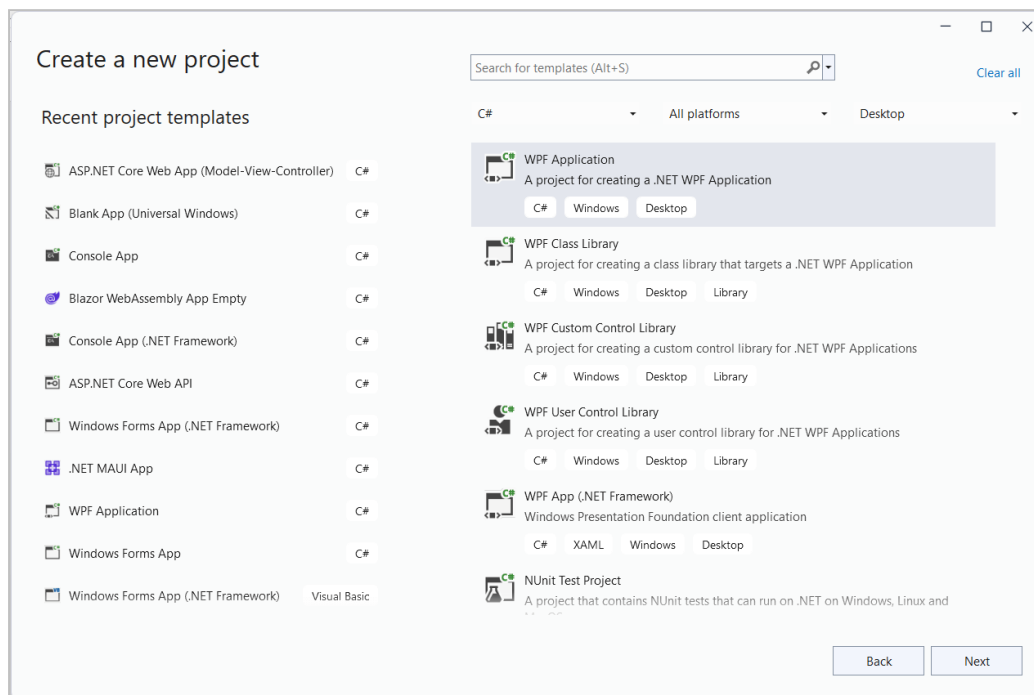
1. Create a new WPF application.
2. Install and configure the Excel-to-PDF converter library.

3. Design the UI for selecting Excel documents.
4. Convert Excel file to PDF and print the document.

Step 1: Create a new WPF app

First, we are going to [create a new WPF application in Visual Studio](#).

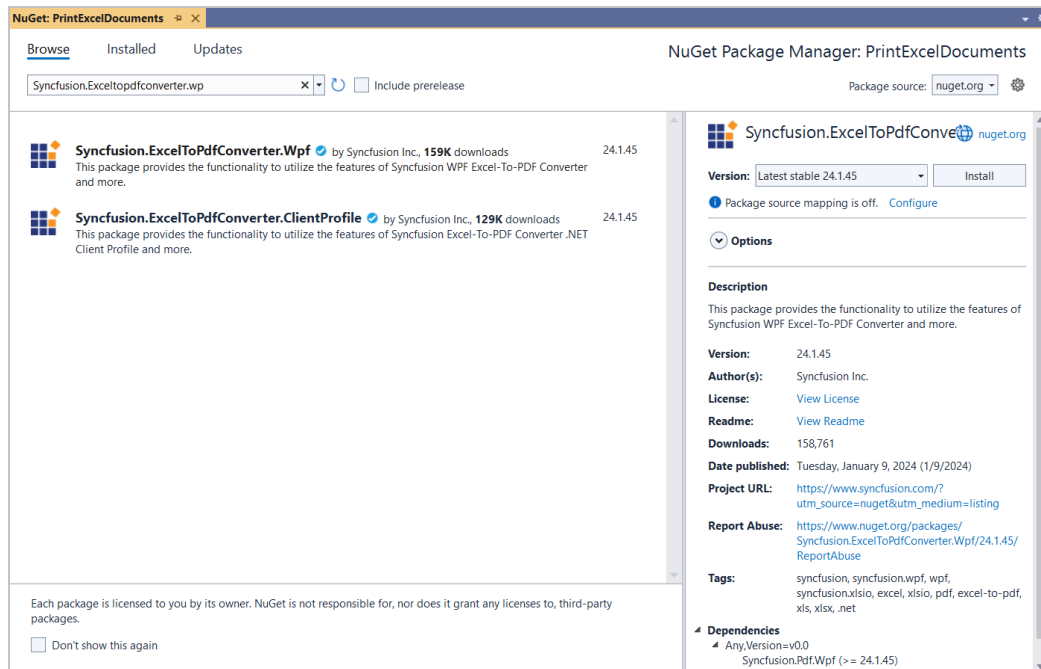
Refer to the following image.



Step 2: Install and configure the Excel-to-PDF converter library

Then, add the following NuGet packages to your WPF app:

- [Syncfusion.ExcelToPdfConverter.Wpf](#)
- [System.Drawing.Common](#)



Handle Excel files like a pro with Syncfusion's C# Excel Library, offering well-documented APIs for each functionality.

[Read Now](#)

Step 3: Design the UI for selecting Excel documents

Add the required UI buttons for selecting Excel documents and performing the print operation in the **MainWindow.xaml** file.

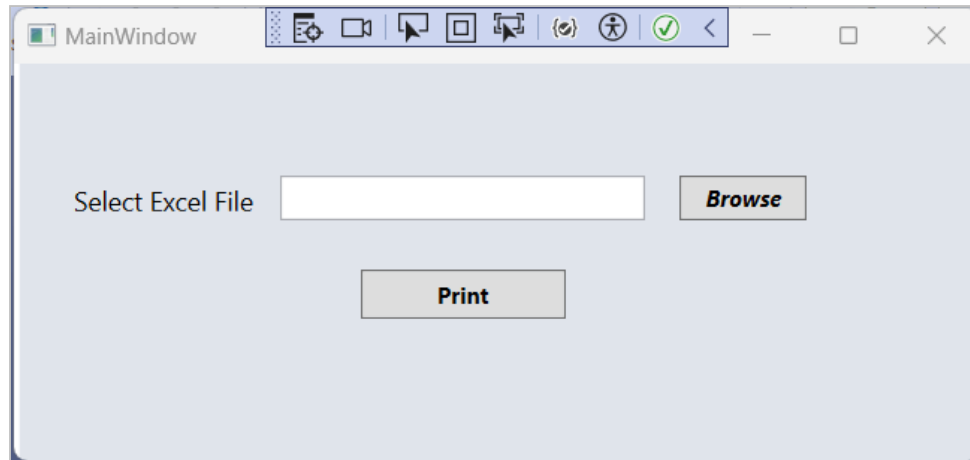
Refer to the following code example.



```
<Window x:Name="Print_Excel" x:Class="PrintExcelDocuments.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:PrintExcelDocuments"
    mc:Ignorable="d"
    Title="MainWindow" Height="250" Width="532">
    <Grid Margin="3,-1,0,-6" Background="#FFE0E4EB">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="37*" />
            <ColumnDefinition Width="21*" />
            <ColumnDefinition Width="81*" />
            <ColumnDefinition Width="68*" />
            <ColumnDefinition Width="278*" />
            <ColumnDefinition Width="44*" />
        </Grid.ColumnDefinitions>
        <Button Click="PrintExcel" Content="Print" HorizontalAlignment="Left"
        <Label Content="Select Excel File" HorizontalAlignment="Left" Margin="24
        <TextBox Name="filePath" HorizontalAlignment="Left" Margin="4,61,0,0" Te
```

```
<Button Content="Browse" HorizontalAlignment="Left" Margin="152,61,0,0"  
</Grid>  
</Window>
```

After executing the previous code example, the UI will look like the following image.



Print window

You can choose the Excel document by clicking **Browse** and then click **Print** to start the printing process.



Witness the possibilities in demos showcasing the robust features of Syncfusion's C# Excel Library.

Try Now

Step 4: Convert Excel file to PDF and print the document

Finally, add the following code in the **MainWindow.xaml.cs** file to convert the Excel document to PDF using the [Syncfusion Excel-to-PDF converter](#) and print the document.

```
using System.Drawing.Printing;
using System.Windows;
using Syncfusion.ExcelToPdfConverter;
using Syncfusion.XlsIO;

namespace PrintExcelDocuments
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }
    }
}
```

Copy


```
}

private void SelectFile(object sender, RoutedEventArgs e)
{
    //Initializes FileSavePicker.
    Microsoft.Win32.OpenFileDialog openFileDialog = new Microsoft.Win32.
    openFileDialog.Filter = "Excel Files|*.xls;*.xlsx;*.xlsm,*.xltn,*.cs
    openFileDialog.Title = "Select a Excel File";
    openFileDialog.ShowDialog();

    //Gets the path of specified file.
    filePath.Text = openFileDialog.FileName;
}

private void PrintExcel(object sender, RoutedEventArgs e)
{
    using (ExcelEngine excelEngine = new ExcelEngine())
    {
        IApplication application = excelEngine.Excel;
        application.DefaultVersion = ExcelVersion.Xlsx;

        //Loads or opens an existing workbook through Open method of IWo
        IWorkbook workbook = application.Workbooks.Open(filePath.Text);

        //Initialize the printer settings.
        PrinterSettings printerSettings = new PrinterSettings();

        //Customizing the printer settings.
        printerSettings.PrinterName = "HP LaserJet Pro MFP M127-M128 PCL
        printerSettings.Copies = 2;
        printerSettings.FromPage = 1;
        printerSettings.ToPage = 3;
        printerSettings.DefaultPageSettings.Color = true;
```

```

printerSettings.Duplex = Duplex.Vertical;
printerSettings.Collate = true;

ExcelToPdfConverter converter = new ExcelToPdfConverter(workbook

converter.Print();
    }
}
}
}

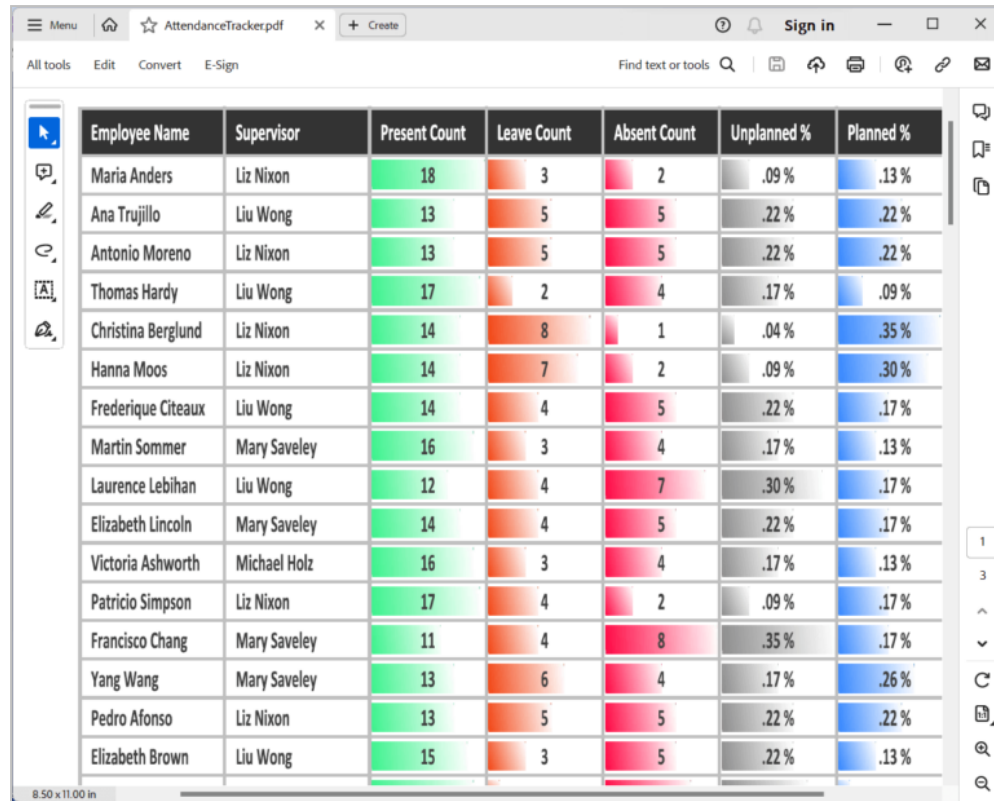
```

The following image shows the input Excel document.

	A	B	C	D	E	F	G
	Employee Name	Supervisor	Present Count	Leave Count	Absent Count	Unplanned %	Planned %
1	Maria Anders	Liz Nixon	18	3	2	8.70 %	13.04 %
2	Ana Trujillo	Liu Wong	13	5	5	21.74 %	21.74 %
3	Antonio Moreno	Liz Nixon	13	5	5	21.74 %	21.74 %
4	Thomas Hardy	Liu Wong	17	2	4	17.39 %	8.70 %
5	Christina Berglund	Liz Nixon	14	8	1	4.35 %	34.78 %
6	Hanna Moos	Liz Nixon	14	7	2	8.70 %	30.43 %
7	Frederique Citeaux	Liu Wong	14	4	5	21.74 %	17.39 %
8	Martin Sommer	Mary Saveley	16	3	4	17.39 %	13.04 %
9	Laurence Lebihan	Liu Wong	12	4	7	30.43 %	17.39 %
10	Elizabeth Lincoln	Mary Saveley	14	4	5	21.74 %	17.39 %
11	Victoria Ashworth	Michael Holz	16	3	4	17.39 %	13.04 %
12	Patricio Simpson	Liz Nixon	17	4	2	8.70 %	17.39 %
13	Francisco Chang	Mary Saveley	11	4	8	34.78 %	17.39 %
14	Yang Wang	Mary Saveley	13	6	4	17.39 %	26.09 %
15	Pedro Afonso	Liz Nixon	13	5	5	21.74 %	21.74 %
16	Elizabeth Brown	Liu Wong	15	3	5	21.74 %	13.04 %
17	Steve Rogers	Michael Holz	16	1	6	26.09 %	4.35 %
18							

Input Excel document

By executing this code example, you will get a printed Excel document as shown in the following image.



Employee Name	Supervisor	Present Count	Leave Count	Absent Count	Unplanned %	Planned %
Maria Anders	Liz Nixon	18	3	2	.09 %	.13 %
Ana Trujillo	Liu Wong	13	5	5	.22 %	.22 %
Antonio Moreno	Liz Nixon	13	5	5	.22 %	.22 %
Thomas Hardy	Liu Wong	17	2	4	.17 %	.09 %
Christina Berglund	Liz Nixon	14	8	1	.04 %	.35 %
Hanna Moos	Liz Nixon	14	7	2	.09 %	.30 %
Frederique Citeaux	Liu Wong	14	4	5	.22 %	.17 %
Martin Sommer	Mary Saveley	16	3	4	.17 %	.13 %
Laurence Lebihan	Liu Wong	12	4	7	.30 %	.17 %
Elizabeth Lincoln	Mary Saveley	14	4	5	.22 %	.17 %
Victoria Ashworth	Michael Holz	16	3	4	.17 %	.13 %
Patricio Simpson	Liz Nixon	17	4	2	.09 %	.17 %
Francisco Chang	Mary Saveley	11	4	8	.35 %	.17 %
Yang Wang	Mary Saveley	13	6	4	.17 %	.26 %
Pedro Afonso	Liz Nixon	13	5	5	.22 %	.22 %
Elizabeth Brown	Liu Wong	15	3	5	.22 %	.13 %

Printed Excel document

References

For more details, refer to the [print Excel documents using C# documentation](#) and [GitHub demos](#).



Trusted by industry giants worldwide, Syncfusion's Excel Framework has a proven track record of reliability and excellence.

Try It Free

Conclusion

Thanks for reading! In this blog, we explored the steps involved in printing Excel documents using C# with the [Syncfusion Excel Library](#). Additionally, we learned how to convert an Excel document to a PDF document in C# using our Syncfusion [Excel-to-PDF converter library](#). With Essential XlsIO, you also have the ability to export Excel data to [images](#), [data tables](#), [CSV](#), [TSV](#), [HTML](#), [collections of objects](#), [ODS](#), [JSON](#), and other file formats.

Feel free to try out these methods and share your feedback in the comments section of this blog post!

For existing customers, the new version of Essential Studio® is available for download from the [License and Downloads](#) page. If you are not yet a Syncfusion customer, you can try our 30-day [free trial](#) to check out our available features.

For questions, you can contact us through our [support forum](#), [support portal](#), or [feedback portal](#). We are happy to assist you!

Related blogs

- [Merge Multiple Excel Files into One in Just 3 Steps Using C#](#)
- [6 Easy Ways to Export Data to Excel in C#](#)
- [Seamlessly Import and Export CSV Data in Excel Using C#](#)
- [How to Export Data from SQL Server to Excel Table in C#](#)
- [Export Data from Collection to Excel and Group It in C#](#)
- [Export Data to a Predefined Excel Template in C#](#)
- [Easy Steps to Export HTML Tables to an Excel in C#](#)



MEET THE AUTHOR

Mohan Chandran

Mohan Chandran is an employee at Syncfusion Software with 4+ years of experience working in an Excel-related library called XlsIO. He is good at finding solutions and resolving queries related to Excel.