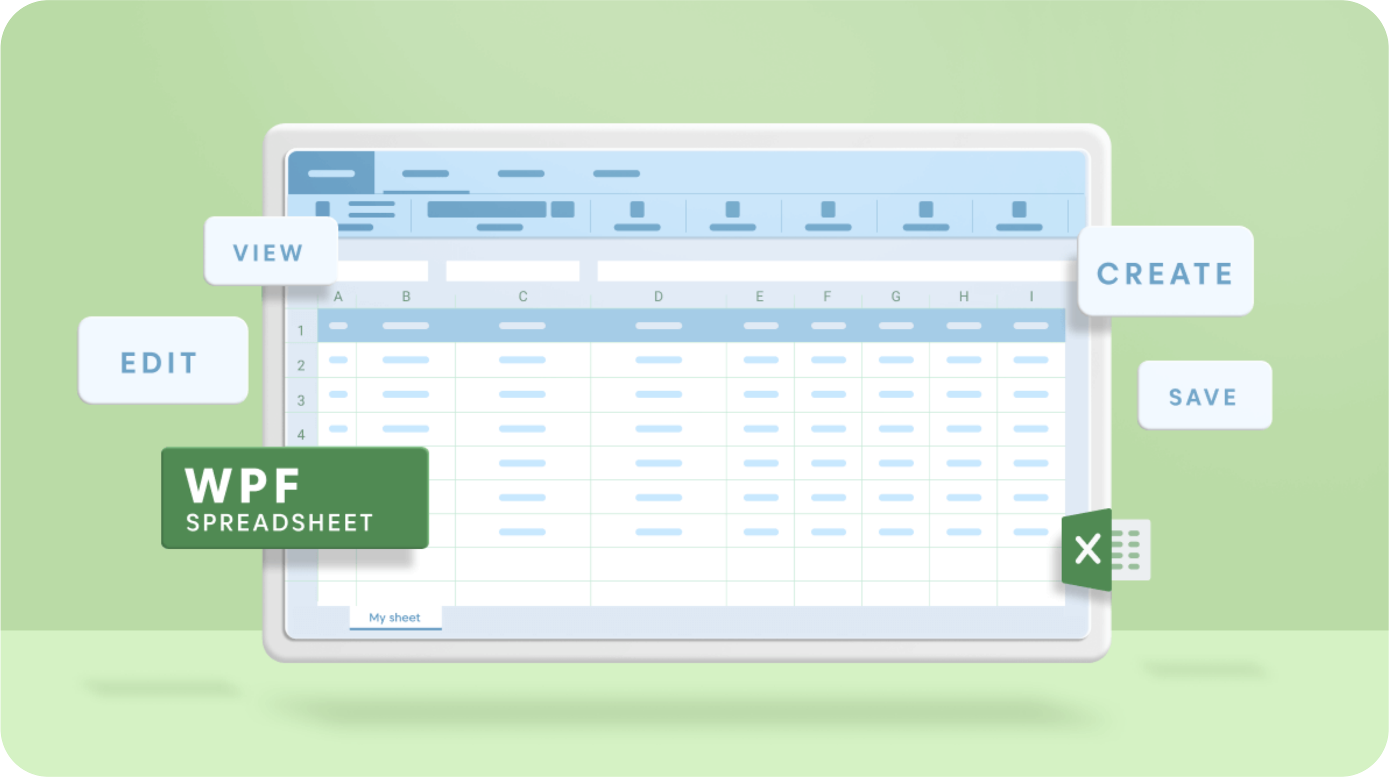
Create, View, Edit, and Save Your Excel Files Using WPF Spreadsheet

Nijamudeen • 5 min read • Nov 19, 2024 • Updated



Spreadsheets are the heart of data analysis and storage. Almost everyone uses spreadsheets at some point, for personal use or for business.

Our Syncfusion WPF Spreadsheet is an Excel-inspired control that allows you to create, view, edit, and format Microsoft Excel files without having Microsoft Excel installed. It provides an integrated ribbon to cover any possible business scenario. In addition, this control supports a built-in calculation engine with more than 400 widely used formulas. It was built with the .NET Excel Library (Essential XlsIO), which features a full-fledged object model similar to the Microsoft Office automation libraries.

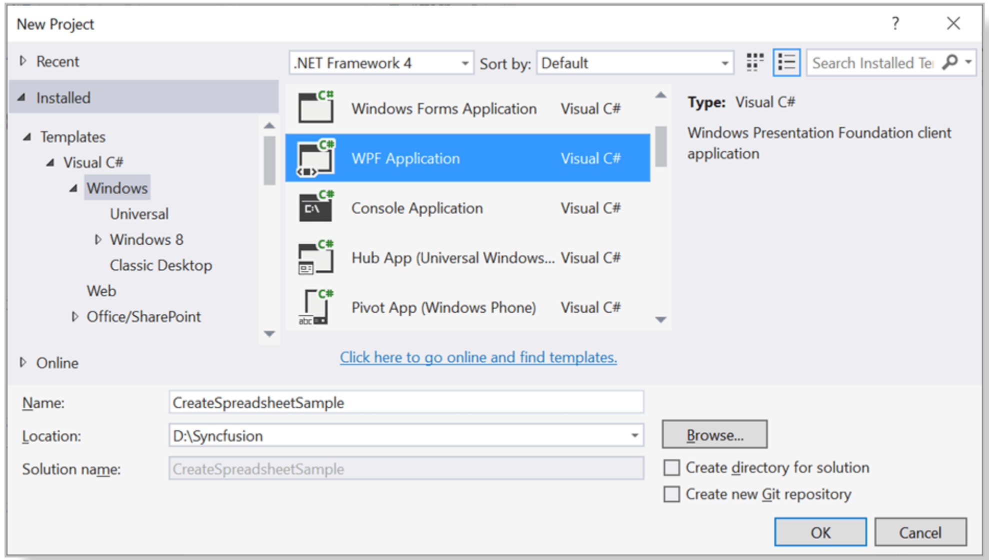
In this blog, I will explain how to create, read, and edit an Excel file using our Syncfusion WPF Spreadsheet control.

Note: If you are new to our Spreadsheet control, then please read the Getting Started with WPF Spreadsheet documentation before proceeding.

Create a simple Spreadsheet project

The following steps illustrate how to create a simple Spreadsheet project in WPF:

1. First, open **Visual Studio** and select **Create a New Project**.
2. Then, the New Project dialog will open. Choose **WPF Application** from the dropdown list and provide a name for your application. Refer to the following screenshot.



New Project

.NET Framework 4

Sort by: Default

WPF Application Visual C#

Type: Visual C#

Windows Presentation Foundation client application

Name: CreateSpreadsheetSample

Location: D:\Syncfusion

Solution name: CreateSpreadsheetSample

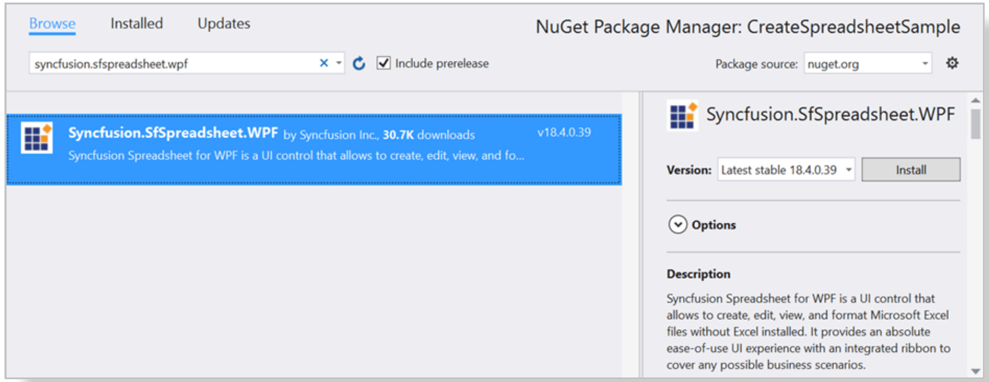
□ Create directory for solution

□ Create new Git repository

?
Recent
Installed
Templates
Visual C#
Windows
Universal
Windows 8
Classic Desktop
Web
Office/SharePoint
Online
Search Installed Te
Windows Forms Application Visual C#
Console Application Visual C#
Hub App (Universal Windows... Visual C#
Pivot App (Windows Phone) Visual C#
Click here to go online and find templates.
Browse...
OK
Cancel

Create a New WPF Project

3. Now, install the Syncfusion.SfSpreadsheet.WPF NuGet package as a reference to your .NET Framework app from Nuget.org.



NuGet Package Manager: CreateSpreadsheetSample

syncfusion.sfspreadsheet.wpf ☑ Include prerelease

Package source: nuget.org

Syncfusion.SfSpreadsheet.WPF by Syncfusion Inc., **30.7K** downloads v18.4.0.39
Syncfusion Spreadsheet for WPF is a UI control that allows to create, edit, view, and fo...

Syncfusion.SfSpreadsheet.WPF

Version: Latest stable 18.4.0.39

Description

Syncfusion Spreadsheet for WPF is a UI control that allows to create, edit, view, and format Microsoft Excel files without Excel installed. It provides an absolute ease-of-use UI experience with an integrated ribbon to cover any possible business scenarios.

Install the Syncfusion.SfSpreadsheet.WPF NuGet Package

The spreadsheet is available in the namespace **Syncfusion.UI.Xaml.Spreadsheet** and you can create it programmatically either by using the XAML or C# code.

4. Next, add the **SpreadsheetRibbon** and bind the Spreadsheet as the DataContext to the SpreadsheetRibbon to make an interaction between the ribbon items and Spreadsheet.

5. To add the SpreadsheetRibbon to your application, use the RibbonWindow since the backstage of the ribbon will open only when the

ribbon is loaded under the RibbonWindow.

Refer to the following code.

Copy

```
<syncfusion:RibbonWindow x:Class="SpreadsheetDemo.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:syncfusion="http://schemas.syncfusion.com/wpf"
mc:Ignorable="d">
<syncfusion:SfSpreadsheetRibbon DataContext= "{Binding ElementName=spreadsh
<syncfusion:SfSpreadsheet x:Name="spreadsheet"/>
</ syncfusion:RibbonWindow >
```

Now, we have created a simple WPF Spreadsheet project.

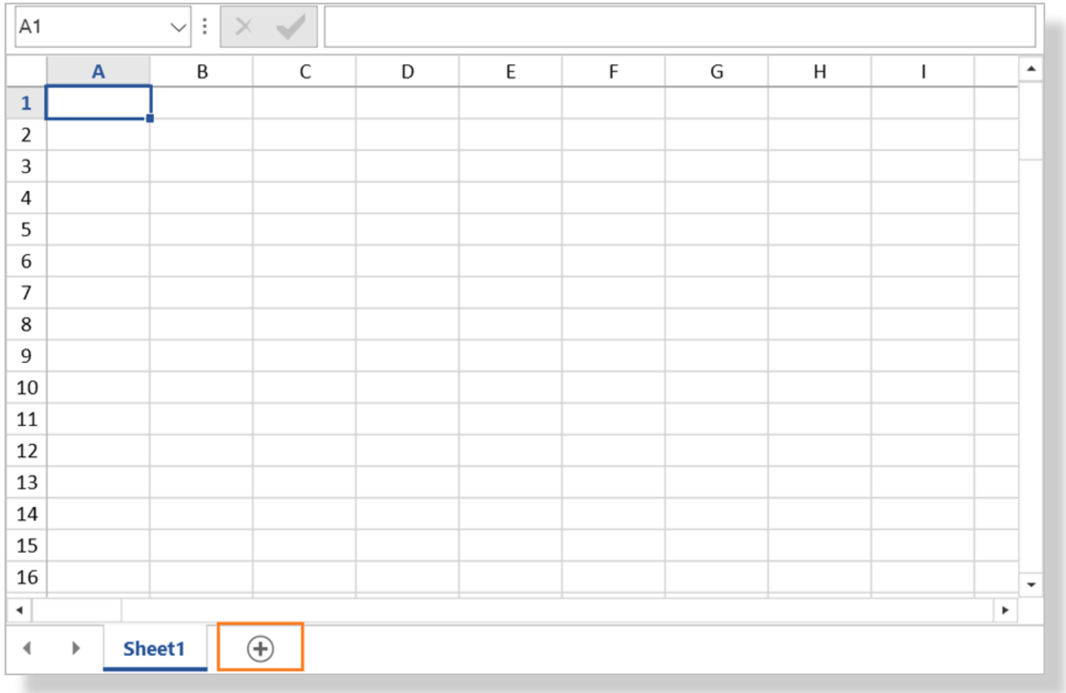
Create a new Excel workbook

You can create a new workbook by using the Create method. By default, a workbook will be created with a single worksheet. Use the following code to load the Spreadsheet workbook with a specified number of worksheets.

Copy

```
spreadsheetControl.Create(2);
```

Also, you can add new worksheets on-demand in the WPF Spreadsheet control by clicking the **AddNew** button. Refer to the following screenshot.



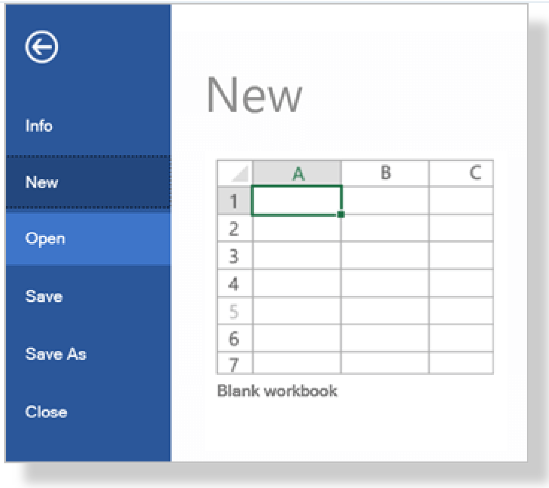
Creating an Excel Workbook Using WPF Spreadsheet

View the existing Excel sheet

The following code snippet illustrates how to view Excel files using the WPF Spreadsheet control.

```
/// View the Excel file.
spreadsheetControl.Open (@"..\..\Data\GettingStarted.xlsx");
                   (or)
ExcelEngine excelEngine = new ExcelEngine();
IWorkbook workbook = excelEngine.Excel.Workbooks.Open(@"..\..\Data\GettingStarte
spreadsheetControl.Open(workbook);
                   (or)
using (FileStream fileStream = new FileStream(@"..\..\Data\ GettingStarted.xlsx”
{
     spreadsheetControl.Open(fileStream);
}
```

Also, you can open the existing Excel files by clicking **Open** in the **SpreadsheetRibbon** backstage, like in the following screenshot.



Opening an Existing Excel File Using WPF Spreadsheet

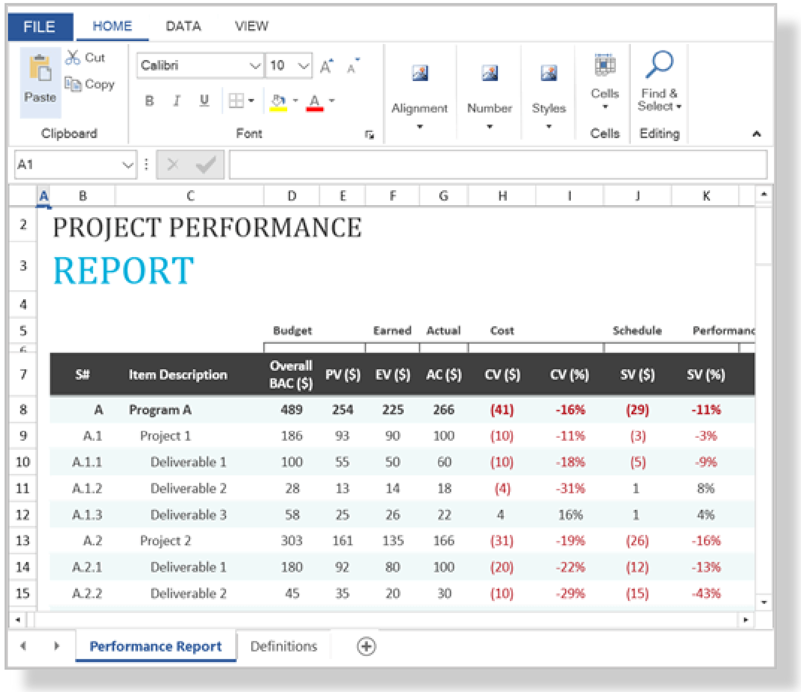
Edit the values in an Excel file

The Spreadsheet provides support for editing, so you can modify and commit the cell values in a workbook. The following code snippet illustrates how to edit data in an Excel file using the WPF Spreadsheet control.

Copy

```
/// Editing a specific cell value.
var range = spreadsheetControl.ActiveSheet.Range[2,2];
spreadsheetControl.ActiveGrid.SetCellValue(range, "Syncfusion");
spreadsheetControl.ActiveGrid.InvalidateCell(2,2);
```

Refer to the following screenshot.



PROJECT PERFORMANCE

REPORT

		Budget		Earned	Actual	Cost		Schedule	Performanc
S#	Item Description	Overall BAC ($)	PV ($)	EV ($)	AC ($)	CV ($)	CV (%)	SV ($)	SV (%)
A	Program A	489	254	225	266	(41)	-16%	(29)	-11%
A.1	Project 1	186	93	90	100	(10)	-11%	(3)	-3%
A.1.1	Deliverable 1	100	55	50	60	(10)	-18%	(5)	-9%
A.1.2	Deliverable 2	28	13	14	18	(4)	-31%	1	8%
A.1.3	Deliverable 3	58	25	26	22	4	16%	1	4%
A.2	Project 2	303	161	135	166	(31)	-19%	(26)	-16%
A.2.1	Deliverable 1	180	92	80	100	(20)	-22%	(12)	-13%
A.2.2	Deliverable 2	45	35	20	30	(10)	-29%	(15)	-43%

Performance Report | Definitions

Editing the Cell Values in an Excel File Using WPF Spreadsheet

Saving an Excel sheet

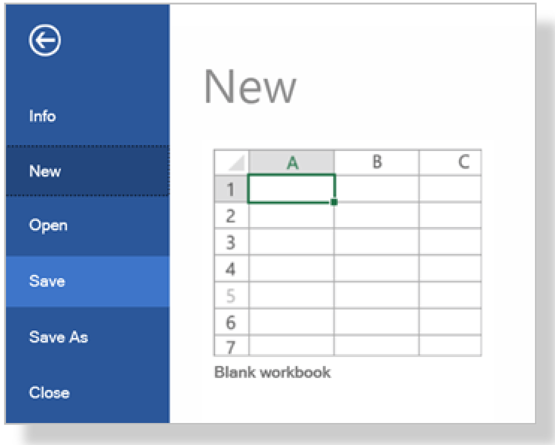
You can save Excel workbooks using the Save() method. If the workbook already exists in the system drive, then it will be saved in the same location. Otherwise, the **Save** dialog box will open to let you save the workbook in a specified location.

Refer to the following code snippet.

```
/// Save the changes made in the file. If the file is not created yet, then it p
spreadsheetControl.Save();

/// Save the changes made in the file using the SaveFileDialog.
spreadsheetControl.SaveAs();
```

Also, you can save the Excel files using **Save** or **SaveAs** in the ribbon backstage.



Saving an Excel File Using WPF Spreadsheet

GitHub reference

You can download the complete sample from Create, View, Edit, and Save Excel Files Using WPF Spreadsheet demo.

Conclusion

Thanks for reading! In this blog post, we gave a quick overview of how to create, view, edit, and save Excel files using the Syncfusion WPF Spreadsheet control. This control also supports hyperlinks, bookmarks, encryption, localization, cell formatting, and more. Try out these versatile features and share your feedback in the comments section of this blog post!

The Spreadsheet control is also available in our ASP.NET (Core, MVC, Web Forms), JavaScript, Angular, React, Vue, UWP, WinForms, and WPF platforms.

If you are an existing Syncfusion user, please download the latest version from the License and Downloads page and try the new features for yourself. Also, our NuGet packages are available on NuGet.org.

If you aren’t a customer yet, you can try our 30-day free trial to check out these features. Also, try our other demos from this GitHub repository.

You can also contact us through our support forums, Direct-Trac, or feedback portal. We are always happy to assist you!

Related blogs

- How to Interact with the WPF Scheduler Using a Keyboard
- Enjoy 30+ New, Elegant Palettes in WPF Theme Studio
- Add, Edit, and Delete Cloud Shapes in PDF Files with Ease in WPF
- Simple Steps to Design a Temperature Monitor Using WPF Radial Gauge

MEET THE AUTHOR

Nijamudeen

Nijamudeen is the product manager for the Syncfusion Xamarin and Flutter platforms, who build highly scalable custom controls that drive Syncfusion's product vision forward.