


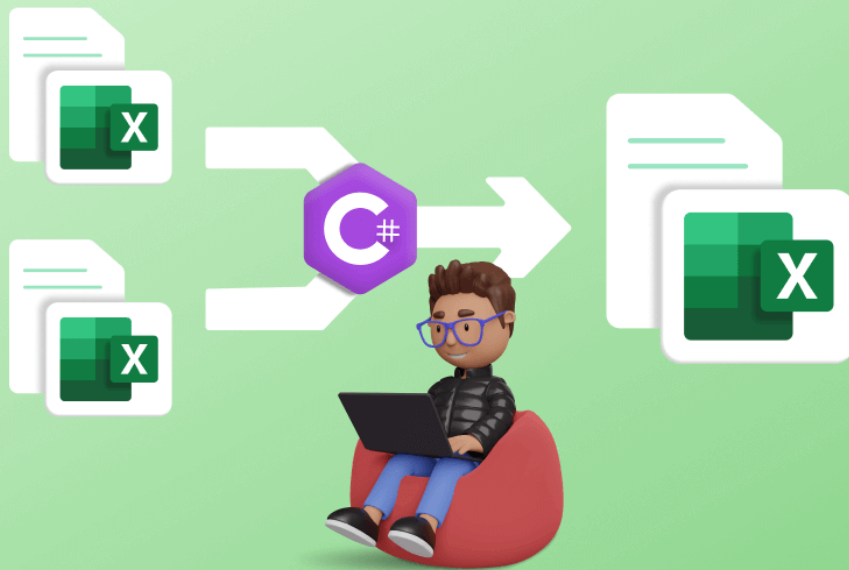


Merge Multiple Excel Files into One in Just 3 Steps Using C#

 **Mohan Chandran** •  4 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

In today's data-centric world, dealing with related data spread over various Excel documents can be challenging. The traditional method of manually merging these documents into a unified one can be time-consuming and susceptible to errors. But what if there was a more efficient solution?

The [Syncfusion Excel Library](#) is a robust tool that facilitates the smooth creation, reading, and editing of Excel documents using C#. This library lets you open and consolidate numerous Excel documents into a single document with just a few lines of code, automating the process, saving valuable time, and ensuring data precision.



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](#)

Whether navigating through many Excel documents, each holding a fragment of a larger whole, or simply seeking a more efficient data management method, this blog is here to assist you. We will guide you through the steps to merge Excel documents using C#, simplifying your data management tasks.

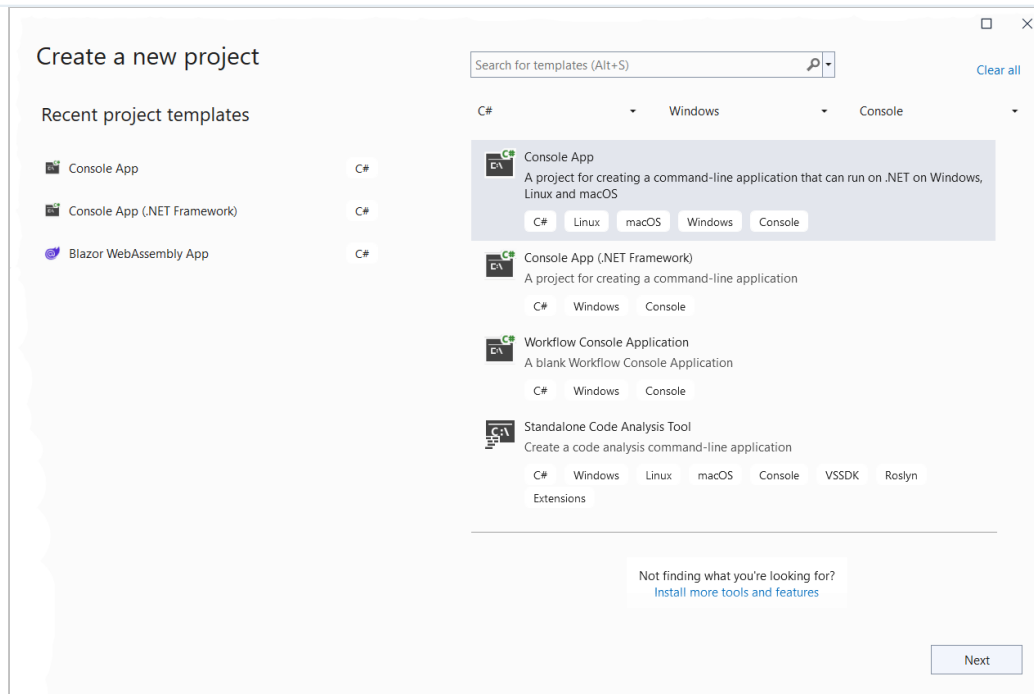
Let's get started!

Note: If you are new to our Excel Library, following our [Getting Started guide](#) is highly recommended.

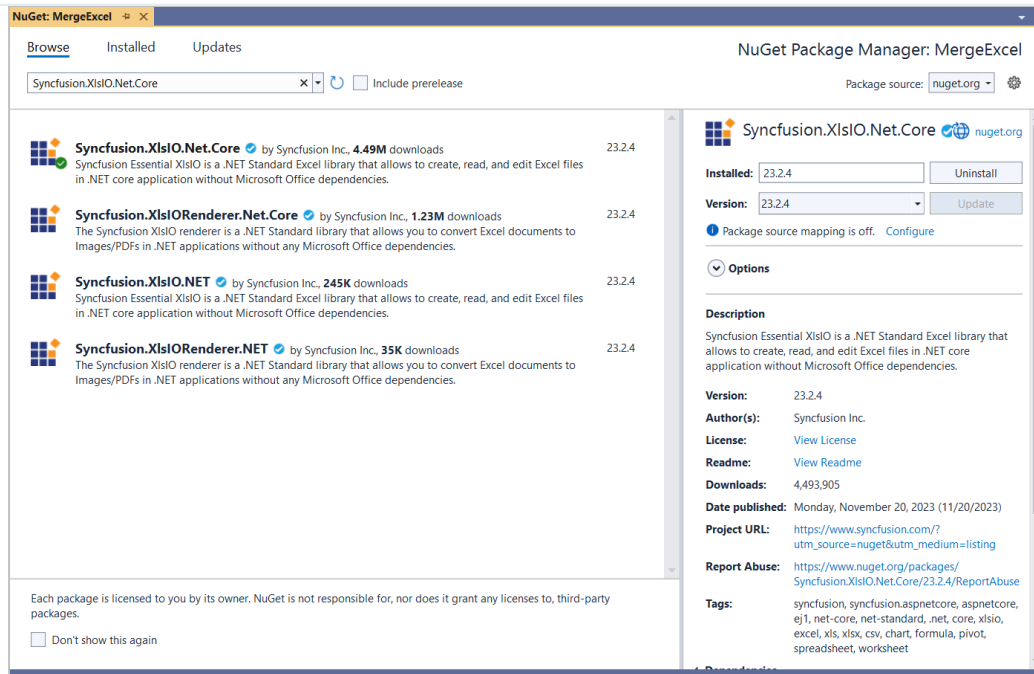
How to merge Excel files with C#

Follow these steps to merge multiple Excel documents into a single file using the Syncfusion Excel Library in C#:

Step 1: First, launch [Visual Studio](#). Then, [create a new .NET Core console application](#).



Step 2: Download and install the most recent version of the [Syncfusion.XlsIO.Net.Core](#) NuGet package.



Step 3: Finally, add the following code to the **Program.cs** file to merge Excel documents from a designated folder path.

```
using Syncfusion.XlsIO;

namespace MergeExcel
{
    class Program
    {
        static void Main(string[] args)
```

Copy

```

{
    string inputPath = @"../../../../../Data/";

    string outputPath = @"../../../../../Output/";

    FileInfo[] files = new DirectoryInfo(inputPath).GetFiles();

    List<Stream> streams = new List<Stream>();

    foreach (FileInfo file in files)
    {
        streams.Add(file.OpenRead());
    }

    Stream mergedStream = MergeExcelDocuments(streams);

    FileStream fileStream = new FileStream(outputPath + "MergedExcel.xls",
        FileMode.Create, FileAccess.Write);
    mergedStream.Position = 0;
    mergedStream.CopyTo(fileStream);
    fileStream.Close();
}

/// <summary>
/// Merge Excel documents from the list of Excel streams.
/// </summary>
/// <param name="streams">List of Excel document stream to be merged</param>
/// <returns></returns>
public static Stream MergeExcelDocuments(List<Stream> streams)
{
    using (ExcelEngine excelEngine = new ExcelEngine())
    {
        IApplication application = excelEngine.Excel;
        application.DefaultVersion = ExcelVersion.Xlsx;
    }
}

```

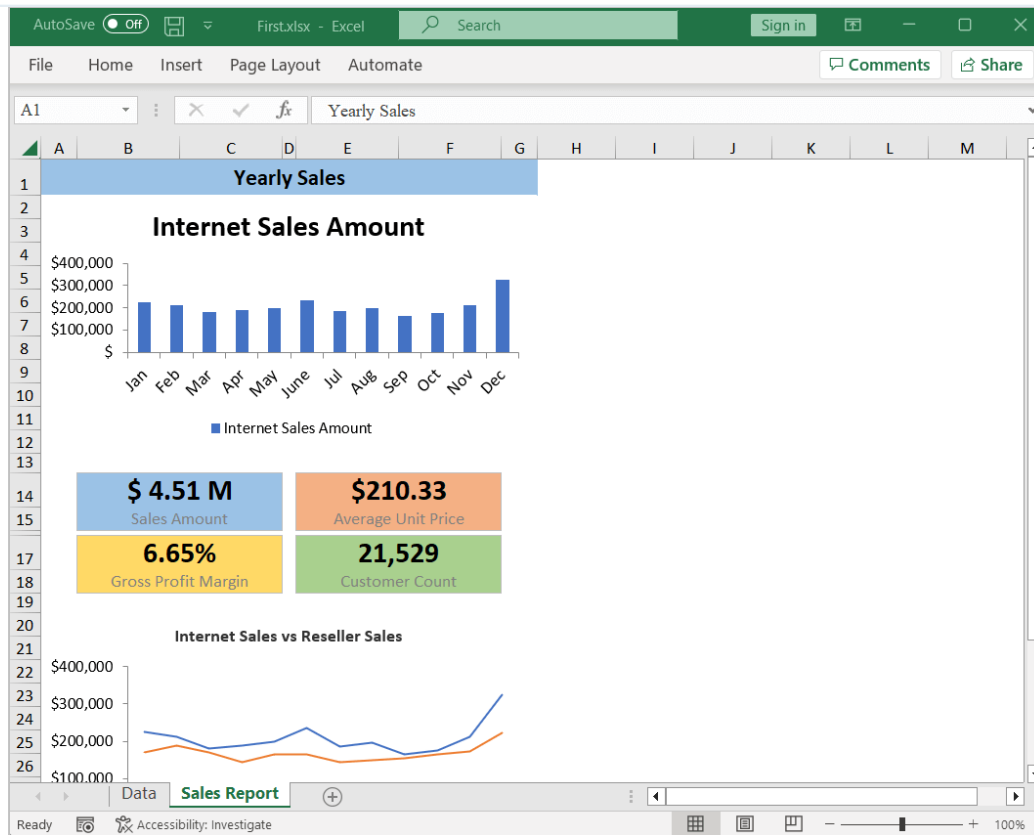
```
        IWorkbook workbook = application.Workbooks.Create(0);







        //Loop through each Excel document and add the worksheets to the
        foreach (Stream stream in streams)
        {
            stream.Position = 0;
            IWorkbook tempWorkbook = application.Workbooks.Open(stream);
            workbook.Worksheets.AddCopy(tempWorkbook.Worksheets);
            tempWorkbook.Close();
        }



        //Save the workbook to a memory stream.
        MemoryStream memoryStream = new MemoryStream();
        workbook.Version = ExcelVersion.Xlsx;
        workbook.SaveAs(memoryStream);




        return memoryStream;
    }
}
```

Refer to the following images showing the input Excel documents.



AutoSave  Off Second.xlsx • Saved to this PC     

File Home Insert Page Layout Automate  Comments  Share


A1   






A B C D E F G H I J K L M

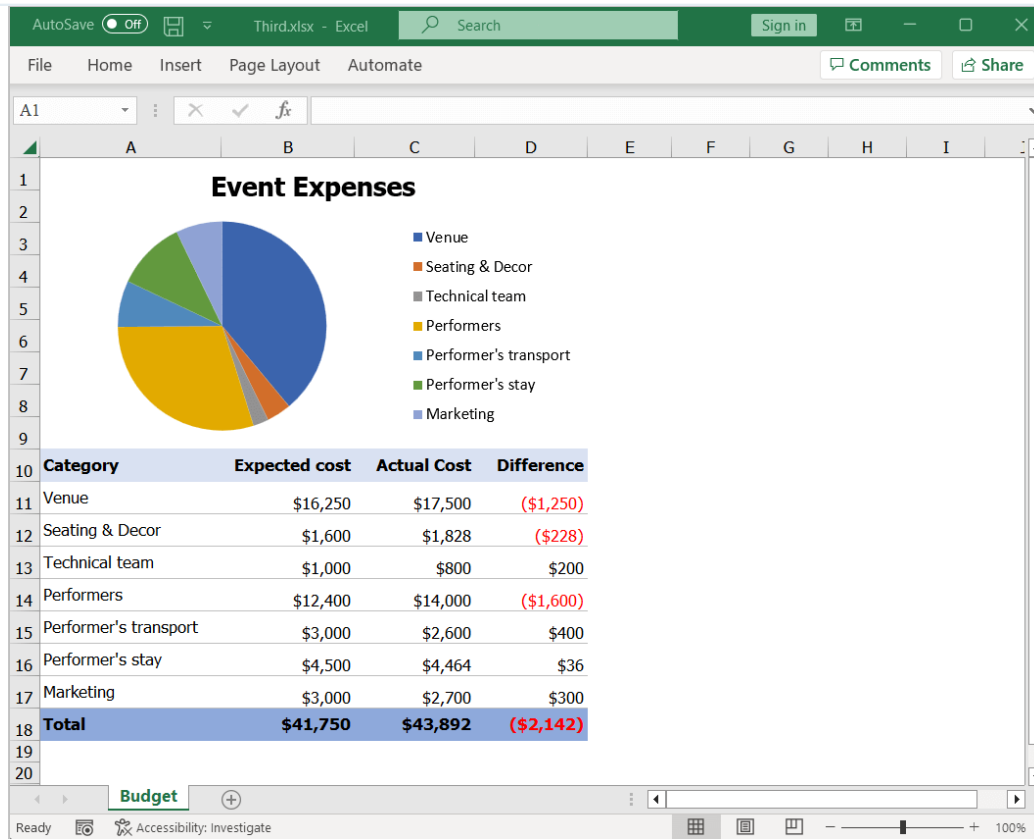
Antony Jose

Name	Antony Jose
Title	Sales Manager
Birth Date	03/08/1963
Hire date	04/01/1992
Home phone	(206) 555-3412
Extension	3355
Home address	722 Moss Bay Blvd
Salary	\$5,000.00

Antony Jose graduated from St. Andrews University, Scotland, with a BSC degree in 1976. Upon joining the company as a sales representative in 1992, he spent 6 months in an orientation program at the Seattle office and then returned to his permanent post in London. He was promoted to sales manager in March 1993.

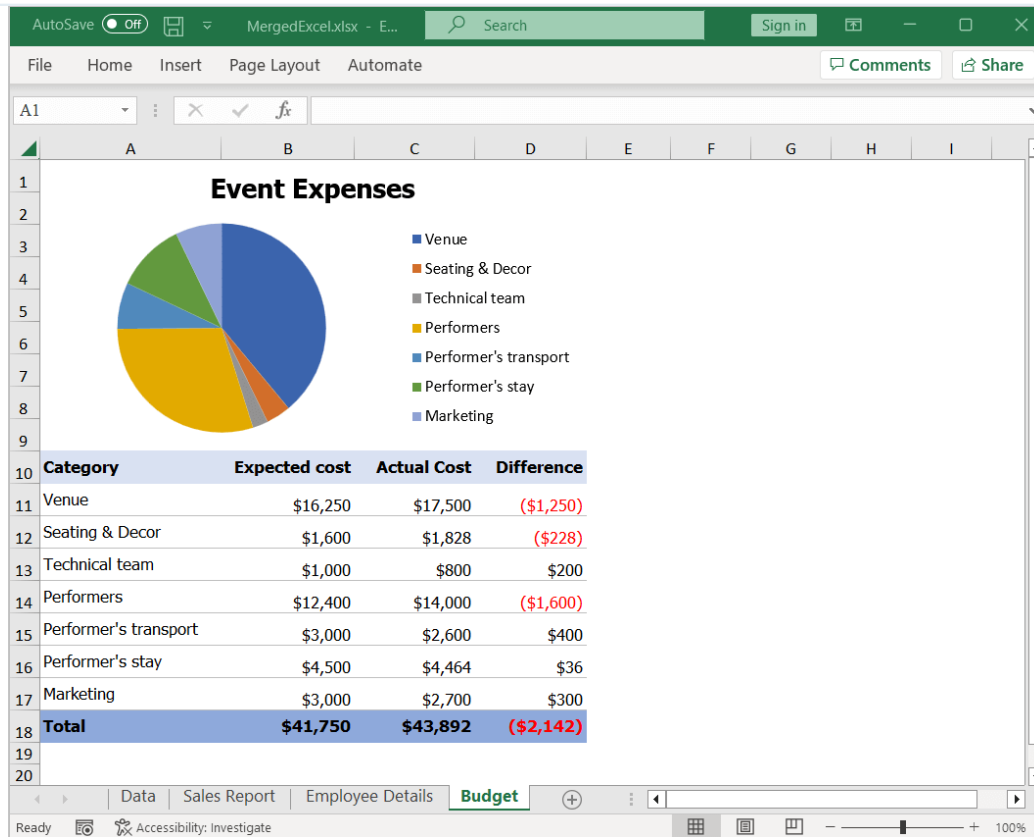
Employee Details 

Ready  Accessibility: Good to go     100%



Input Excel documents

After executing the previous code example, the output will look like the following image. See the number of Excel sheets added at the bottom of the Excel file.



Merging multiple Excel documents with the Syncfusion Excel Library using C#

GitHub reference

You can download the example for merging Excel documents in C# on [this GitHub page](#).



Don't settle for ordinary spreadsheet solutions. Switch to Syncfusion and upgrade the way you handle Excel files in your apps!

Try It Free

Wrapping up

Thanks for reading! As you can see, the [Syncfusion Excel \(XlsIO\) Library](#) lets you effortlessly merge multiple Excel documents into a single document in C# with just three simple steps. Take a moment to peruse the Excel Library [documentation](#), where you'll find other importing options and features like [data tables](#), [collection objects](#), [grid view](#), [data columns](#), and [HTML](#), all with accompanying code samples.

Using the Excel Library, you can export Excel data to [PDFs](#), [images](#), [data tables](#), [CSV](#), [TSV](#), [HTML](#), [collections of objects](#), [ODS](#), [JSON](#), and other file formats.

Are you already a Syncfusion user? You can download the product setup after [logging in](#). If you're not yet a Syncfusion user, you can download a free 30-day trial [from this page](#).

Please let us know in the comments section below if you have any questions about these features. You can also contact us through our [support forum](#), [support portal](#), or [feedback portal](#). We are always happy to assist you!

Related blogs

- [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 Tables in C#](#)
- [Export Data from Collections 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 Worksheet 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.