Read excel file in C# (.XLSX or .XLS using OLEDB / EPPlus or Interop)

👁: 15383

Last Updated : 15/04/2021

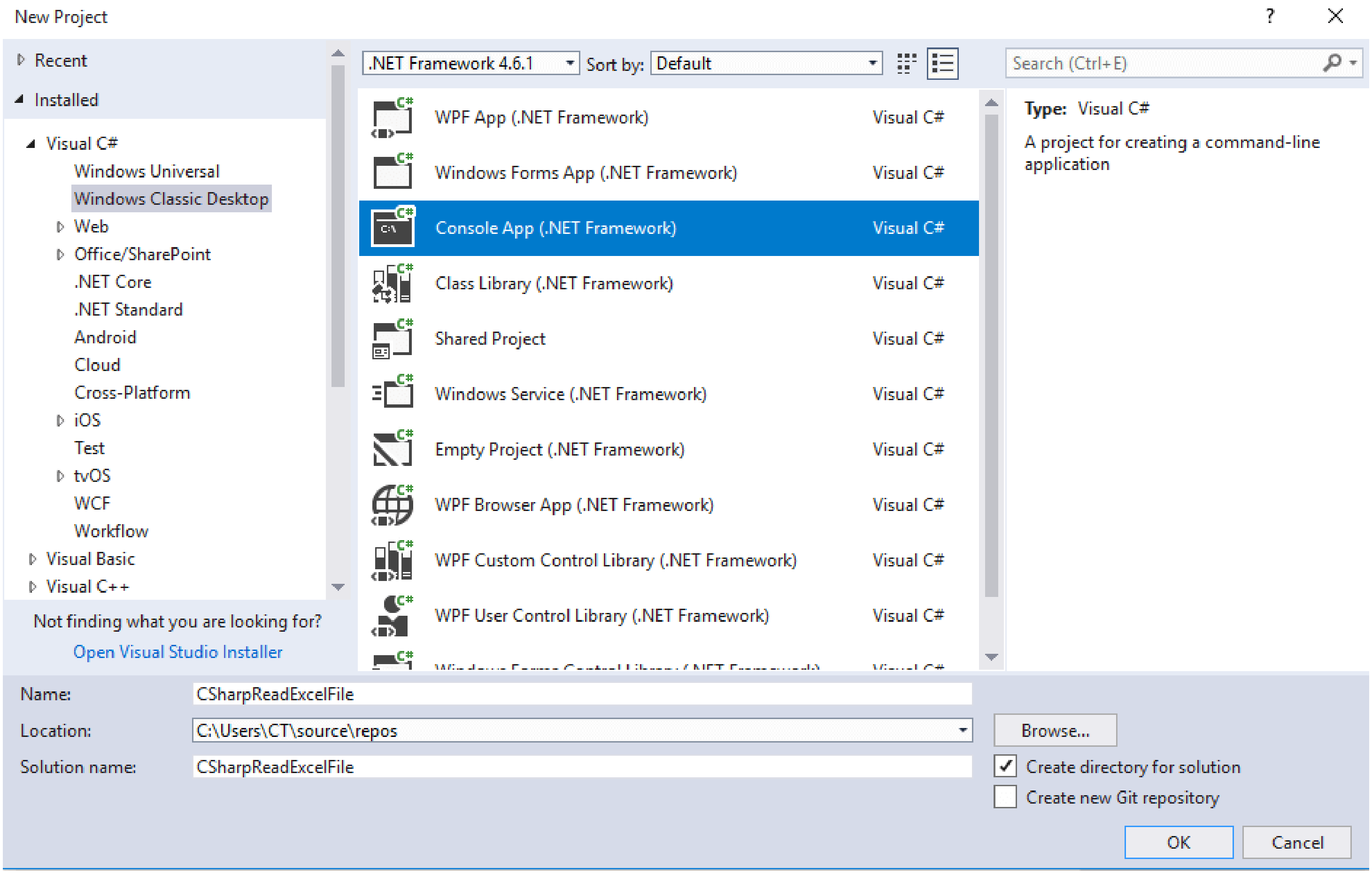
Posted By :- Vinnu (/users/profiles/4)

Whether you are working on Windows or on Web or on Console application, at one stage we need to process data and excel file is widely used for it, so in previous article, we have explained about creating excel in C# without interop (/article/asp-net/create-excel-file-using-c-without-using-office-or-interop/138) but in this article, I am going to provide you code to open and read excel file (.xls or .xlsx) in C# line by line in Console application using OLEDB or EPPlus or Interop (all 3 methods), you can use the same code C# code to fill ASP.NET GridView or MVC application table.

So, let's get started with it.

Step 1: Create a new Console applicaiton in your Visual Studio, by navigating to File->New->Project-> Select "Windows Classic dekstop" from left-pane & "Console-App" from right-pane -> Provide a name to your application "CSharpReadExcelFile" -> Click "OK"

X


New Project
Recent
Installed
Visual C#
Windows Universal
Windows Classic Desktop
Web
Office/SharePoint
.NET Core
.NET Standard
Android
Cloud
Cross-Platform
iOS
Test
tvOS
WCF
Workflow
Visual Basic
Visual C++
Not finding what you are looking for?
Open Visual Studio Installer
.NET Framework 4.6.1
Sort by: Default
Search (Ctrl+E)
WPF App (.NET Framework) Visual C#
Windows Forms App (.NET Framework) Visual C#
Console App (.NET Framework) Visual C#
Class Library (.NET Framework) Visual C#
Shared Project Visual C#
Windows Service (.NET Framework) Visual C#
Empty Project (.NET Framework) Visual C#
WPF Browser App (.NET Framework) Visual C#
WPF Custom Control Library (.NET Framework) Visual C#
WPF User Control Library (.NET Framework) Visual C#
Browse...
OK
Cancel

Type: Visual C#

A project for creating a command-line application

Name: CSharpReadExcelFile

Location: C:\Users\CT\source\repos

Solution name: CSharpReadExcelFile

☑ Create directory for solution

□ Create new Git repository

Step 2: Now we have our Console application and we need to add C# code using OLEDB to read excel file, for that we would need connection string with the source URL of excel file.

In the given example as I am using .XLS excel file, here is my connection string

For Excel 97-2003 Format we can use “Microsoft Jet OLEDB Driver 4.0”, while for the Connection String for Excel 2007 Format (.XLSX), we can use “Microsoft Ace OLEDB Driver 12.0” and it's connection string would be as below

```
 string connString = "Provider= Microsoft.ACE.OLEDB.12.0;" + "Data Source=Sample1.xlsx" + ";Extended Properties='Excel 8.0;HDR=Yes'";
```

In the above Connection string's you may see extended properties HDR=Yes & HDR =No

Use HDR=YES if first excel row contains headers, alternatively,use HDR=NO when your excel's first row is not headers and it's data.

Now, we have connection string , we need to create connection using OLEDB and open it

```
             // Create the connection object
            OleDbConnection oledbConn = new OleDbConnection(connString);

            // Open connection
            oledbConn.Open();
```

Read the excel file using OLEDB connection and fill it in dataset

```
  //here sheet name is Sample-spreadsheet-file, usually it is Sheet1, Sheet2 etc..
                OleDbCommand cmd = new OleDbCommand("SELECT * FROM [Sample-spreadsheet-file$]", oledbConn);


                // Create new OleDbDataAdapter
                OleDbDataAdapter oleda = new OleDbDataAdapter();

                oleda.SelectCommand = cmd;

                // Create a DataSet which will hold the data extracted from the worksheet.
                DataSet ds = new DataSet();

                // Fill the DataSet from the data extracted from the worksheet.
                oleda.Fill(ds, "Employees");
```

in the above code Sample-spreadsheet-file is the name of Sheet.

> Note: With the help of sheet name, you can refer to Excel data, you need to use '$' with sheet name, e.g. Select * from [Sheet1$]

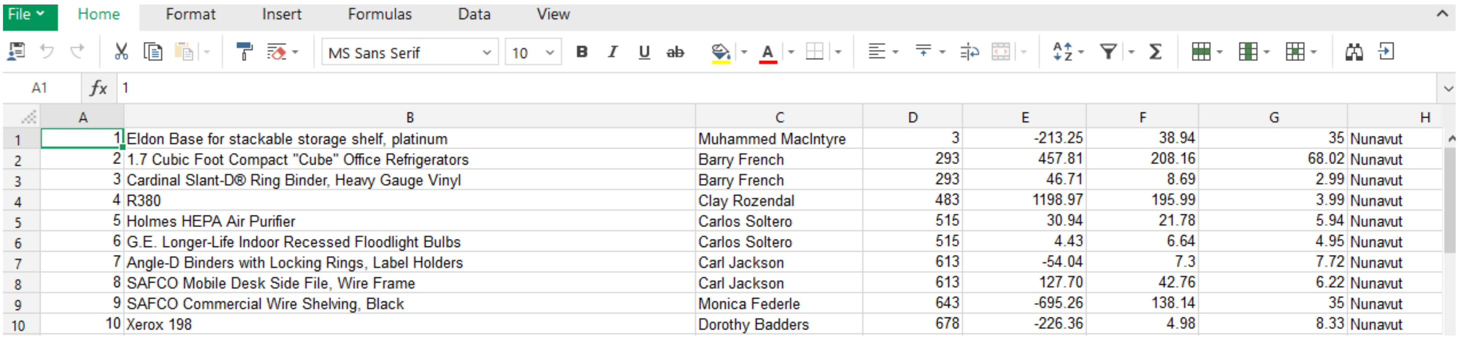
Now loop through each row of excel sheet and print it in Console app

```
 //loop through each row
                foreach(var m in ds.Tables[0].DefaultView)
                {
                    Console.WriteLine(((System.Data.DataRowView)m).Row.ItemArray[0] +" "+((System.Data.DataRowView)m).Row.ItemArray[1] +" "+((System.Data.DataRowView)m).Row.ItemArray[2]);

                }
```

If you are using ASP.NET, you can bind it with Grid View using below code instead of printing it

```
            // Bind the data to the GridView
            GridView1.DataSource = ds.Tables[0].DefaultView;
            GridView1.DataBind();
```

Now, we have discussed each step, suppose this is out Excel file

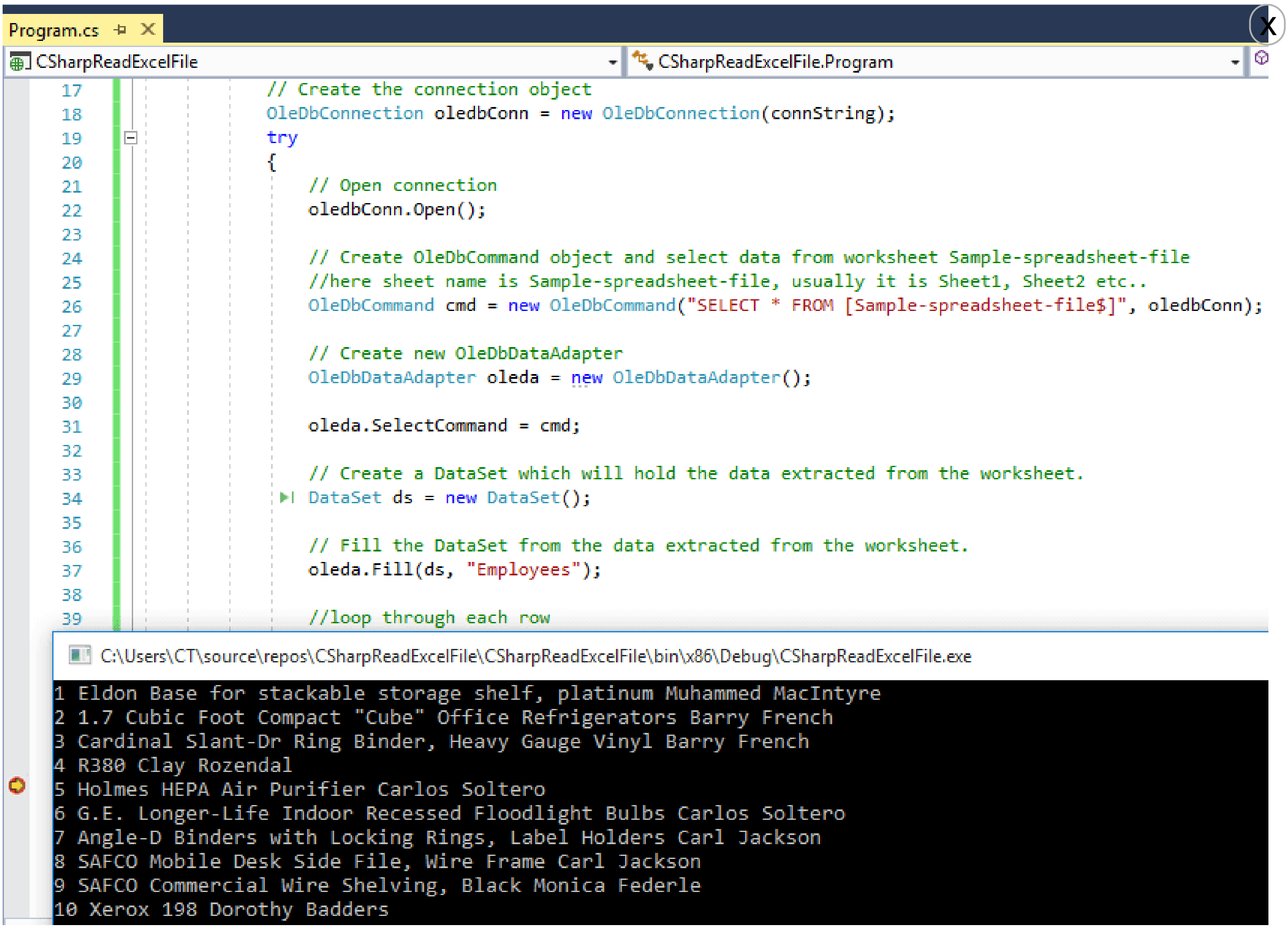


	A	B	C	D	E	F	G	H
1	1	Eldon Base for stackable storage shelf, platinum	Muhammed MacIntyre	3	-213.25	38.94	35	Nunavut
2	2	1.7 Cubic Foot Compact "Cube" Office Refrigerators	Barry French	293	457.81	208.16	68.02	Nunavut
3	3	Cardinal Slant-D® Ring Binder, Heavy Gauge Vinyl	Barry French	293	46.71	8.69	2.99	Nunavut
4	4	R380	Clay Rozendal	483	1198.97	195.99	3.99	Nunavut
5	5	Holmes HEPA Air Purifier	Carlos Soltero	515	30.94	21.78	5.94	Nunavut
6	6	G.E. Longer-Life Indoor Recessed Floodlight Bulbs	Carlos Soltero	515	4.43	6.64	4.95	Nunavut
7	7	Angle-D Binders with Locking Rings, Label Holders	Carl Jackson	613	-54.04	7.3	7.72	Nunavut
8	8	SAFCO Mobile Desk Side File, Wire Frame	Carl Jackson	613	127.70	42.76	6.22	Nunavut
9	9	SAFCO Commercial Wire Shelving, Black	Monica Federle	643	-695.26	138.14	35	Nunavut
10	10	Xerox 198	Dorothy Badders	678	-226.36	4.98	8.33	Nunavut

and ese the code below in your Console application, build and execute it.

```
using System;
using System.Data;
using System.Data.OleDb;



namespace CSharpReadExcelFile
{
    class Program
    {
        static void Main(string[] args)
        {
            //this is the connection string which has OLDB 4.0 Connection and Source URL of file
            //use HDR=YES if first excel row contains headers, HDR=NO means your excel's first row is not headers and it's data.
            string connString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=D:\\Sample1.xls; Extended Properties='Excel 8.0;HDR=NO;IMEX=1;'";



            // Create the connection object
            OleDbConnection oledbConn = new OleDbConnection(connString);
            try
            {
                // Open connection
                oledbConn.Open();

                // Create OleDbCommand object and select data from worksheet Sample-spreadsheet-file
                //here sheet name is Sample-spreadsheet-file, usually it is Sheet1, Sheet2 etc..
                OleDbCommand cmd = new OleDbCommand("SELECT * FROM [Sample-spreadsheet-file$]", oledbConn);


                // Create new OleDbDataAdapter
                OleDbDataAdapter oleda = new OleDbDataAdapter();

                oleda.SelectCommand = cmd;

                // Create a DataSet which will hold the data extracted from the worksheet.
                DataSet ds = new DataSet();

                // Fill the DataSet from the data extracted from the worksheet.
                {
                    Console.WriteLine(((System.Data.DataRowView)m).Row.ItemArray[0] +" "+((System.Data.DataRowView)m).Row.ItemArray[1] +" "+((System.Data.DataRowView)m).Row.ItemArray[2]);

                }

            }
            catch (Exception e)
            {
                Console.WriteLine("Error :" + e.Message);
            }
            finally
            {
                // Close connection
                oledbConn.Close();
            }
        }
    }
}
```

Output of the above code will be as below


Program.cs
CSharpReadExcelFile
CSharpReadExcelFile.Program
C:\Users\CT\source\repos\CSharpReadExcelFile\CSharpReadExcelFile\bin\x86\Debug\CSharpReadExcelFile.exe

```
// Create the connection object
OleDbConnection oledbConn = new OleDbConnection(connString);
try
{
    // Open connection
    oledbConn.Open();

    // Create OleDbCommand object and select data from worksheet Sample-spreadsheet-file
    //here sheet name is Sample-spreadsheet-file, usually it is Sheet1, Sheet2 etc..
    OleDbCommand cmd = new OleDbCommand("SELECT * FROM [Sample-spreadsheet-file$]", oledbConn);

    // Create new OleDbDataAdapter
    OleDbDataAdapter oleda = new OleDbDataAdapter();

    oleda.SelectCommand = cmd;

    // Create a DataSet which will hold the data extracted from the worksheet.
    DataSet ds = new DataSet();

    // Fill the DataSet from the data extracted from the worksheet.
    oleda.Fill(ds, "Employees");

    //loop through each row
```

```
1 Eldon Base for stackable storage shelf, platinum Muhammed MacIntyre
2 1.7 Cubic Foot Compact "Cube" Office Refrigerators Barry French
3 Cardinal Slant-Dr Ring Binder, Heavy Gauge Vinyl Barry French
4 R380 Clay Rozendal
5 Holmes HEPA Air Purifier Carlos Soltero
6 G.E. Longer-Life Indoor Recessed Floodlight Bulbs Carlos Soltero
7 Angle-D Binders with Locking Rings, Label Holders Carl Jackson
8 SAFCO Mobile Desk Side File, Wire Frame Carl Jackson
9 SAFCO Commercial Wire Shelving, Black Monica Federle
10 Xerox 198 Dorothy Badders
```

Now, if you are working on **64 bit operating system**, you may get this error "**The 'Microsoft.Jet.OLEDB.4.0' provider is not registered on the local machine.**".

Resolving this error: If your application is Desktop based, compile your EXE with x86 CPU (Menu Tools, Options, select Projects And Solutions, check the show advanced build configurations. Now in the Build Menu you will be able to go to the Config Manager and set output to x86.)

If **your application is web based, then Enable '32-Bit Applications' in application pool**.

On IIS, change the "**Enable 32-bit Applications**" setting to True, in the Advanced Settings for the Application Pool.

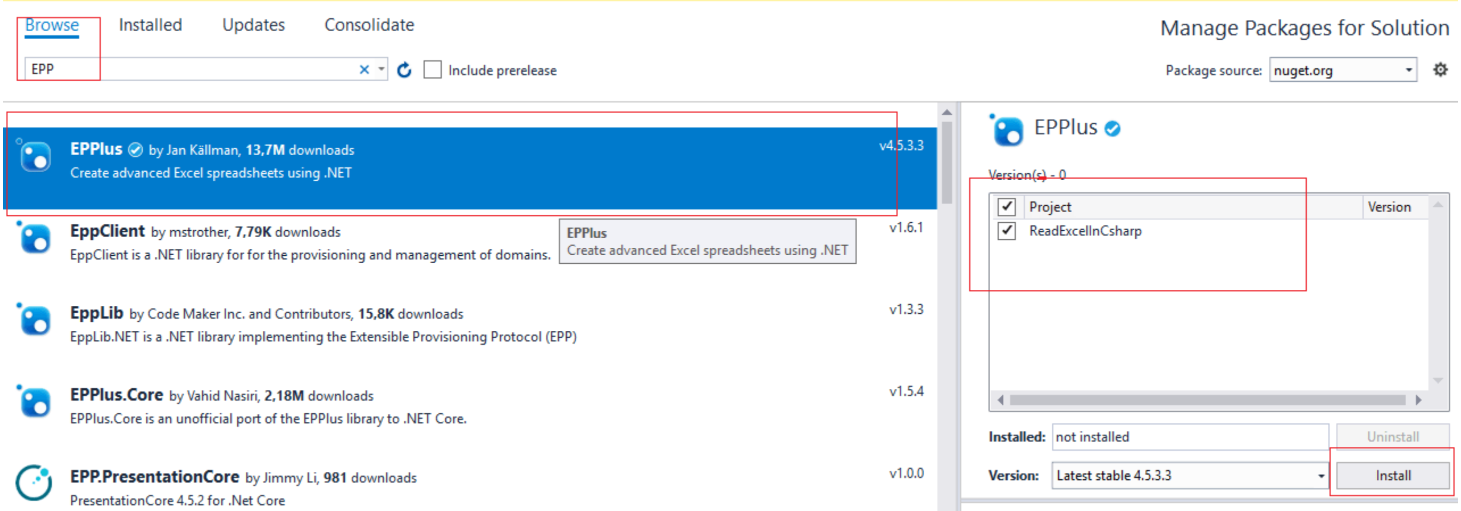
Disadvantage of using OLEDB for Excel

With OLEDB, you cannot format data that you inserted/updated in EXCEL sheet but Interop can do it efficiently. You cannot perform any mathematical operation or working on graphs using OLEDB, but it is really a good way to insert/update data in EXCEL where no Excel application is installed.

Reading Excel file using EPPlus

If you don't want to use OleDb, you can try using EPPlus Nuget package based solution.

For this, you would have to install EPPlus, so navigate to "Tools"-> "Nuget package manager"-> "Manage Nuget for this solution" -> Select "Browse" tab and search for "EPPlus", then install the nuget package.



EPPlus by Jan Källman, **13,7M** downloads v4.5.3.3
Create advanced Excel spreadsheets using .NET

EppClient by mstrother, **7,79K** downloads v1.6.1
EppClient is a .NET library for for the provisioning and management of domains.

EPPlus
Create advanced Excel spreadsheets using .NET

EppLib by Code Maker Inc. and Contributors, **15,8K** downloads v1.3.3
EppLib.NET is a .NET library implementing the Extensible Provisioning Protocol (EPP)

EPPlus.Core by Vahid Nasiri, **2,18M** downloads v1.5.4
EPPlus.Core is an unofficial port of the EPPlus library to .NET Core.

EPP.PresentationCore by Jimmy Li, **981** downloads v1.0.0
PresentationCore 4.5.2 for .Net Core

EPPlus

Version(s) - 0

	Project	Version
☑	ReadExcelInCsharp	

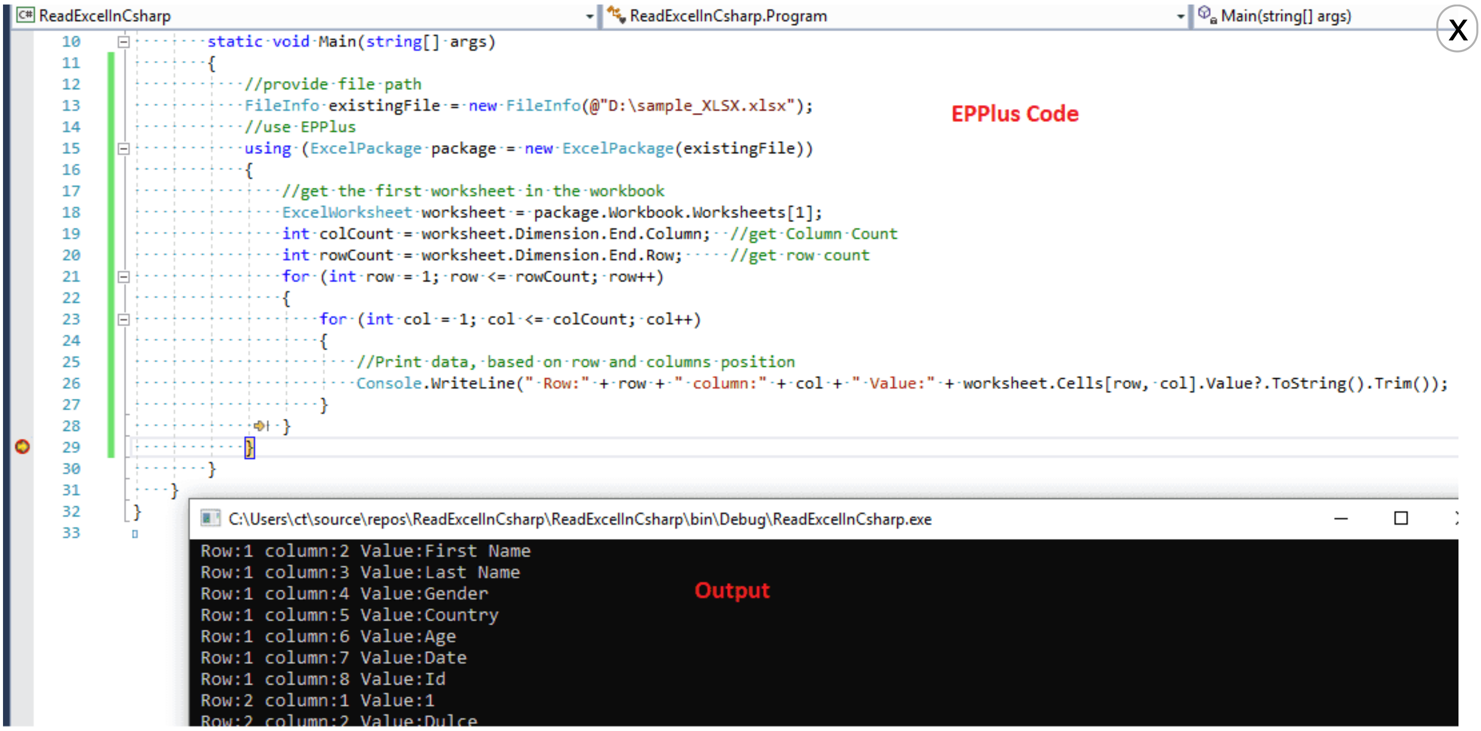
Installed: not installed

Version: Latest stable 4.5.3.3

Once you have installed the package, in your Console application "Program.cs", you can use the code below

```
using OfficeOpenXml;
using System;
using System.IO;



namespace ReadExcelInCsharp
{
    class Program
    {
        static void Main(string[] args)
        {
            //provide file path
            FileInfo existingFile = new FileInfo(@"D:\sample_XLSX.xlsx");
            //use EPPlus
            using (ExcelPackage package = new ExcelPackage(existingFile))
            {
                //get the first worksheet in the workbook
                ExcelWorksheet worksheet = package.Workbook.Worksheets[1];
                int colCount = worksheet.Dimension.End.Column;  //get Column Count
                int rowCount = worksheet.Dimension.End.Row;     //get row count
                for (int row = 1; row <= rowCount; row++)
                {
                    for (int col = 1; col <= colCount; col++)
                    {
                        //Print data, based on row and columns position
                        Console.WriteLine(" Row:" + row + " column:" + col + " Value:" + worksheet.Cells[row, col].Value?.ToString().Trim());
                    }
                }
            }
        }
    }
}
```

Here is the image, which shows console application output with **sample excel file (.xlsx), we are using .xlsx file here for reading in C# using EPPlus**

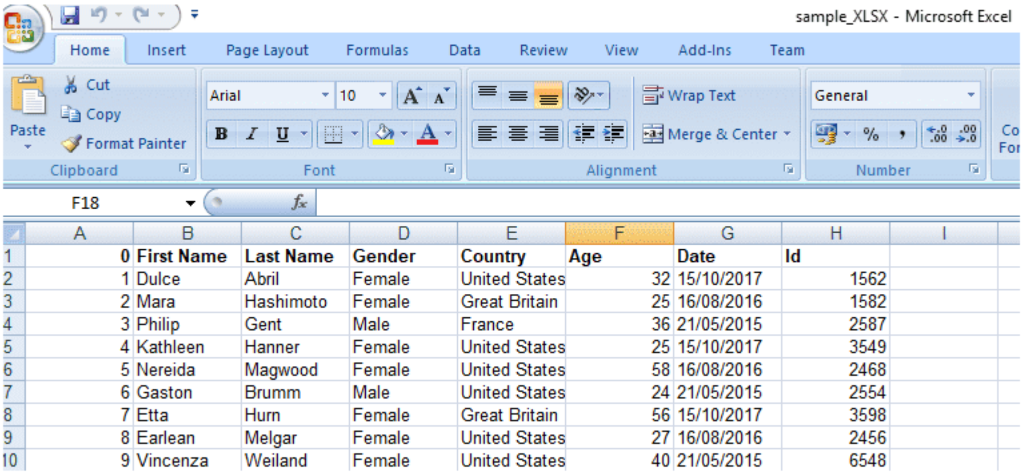

ReadExcelInCsharp
ReadExcelInCsharp.Program
Main(string[] args)
C:\Users\ct\source\repos\ReadExcelInCsharp\ReadExcelInCsharp\bin\Debug\ReadExcelInCsharp.exe

EPPlus Code

```
        static void Main(string[] args)
        {
            //provide file path
            FileInfo existingFile = new FileInfo(@"D:\sample_XSLX.xlsx");
            //use EPPlus
            using (ExcelPackage package = new ExcelPackage(existingFile))
            {
                //get the first worksheet in the workbook
                ExcelWorksheet worksheet = package.Workbook.Worksheets[1];
                int colCount = worksheet.Dimension.End.Column;  //get Column Count
                int rowCount = worksheet.Dimension.End.Row;     //get row count
                for (int row = 1; row <= rowCount; row++)
                {
                    for (int col = 1; col <= colCount; col++)
                    {
                        //Print data, based on row and columns position
                        Console.WriteLine(" Row:" + row + " column:" + col + " Value:" + worksheet.Cells[row, col].Value?.ToString().Trim());
                    }
                }
            }
        }
    }
}
```

Output

```
Row:1 column:2 Value:First Name
Row:1 column:3 Value:Last Name
Row:1 column:4 Value:Gender
Row:1 column:5 Value:Country
Row:1 column:6 Value:Age
Row:1 column:7 Value:Date
Row:1 column:8 Value:Id
Row:2 column:1 Value:1
Row:2 column:2 Value:Dulce
```



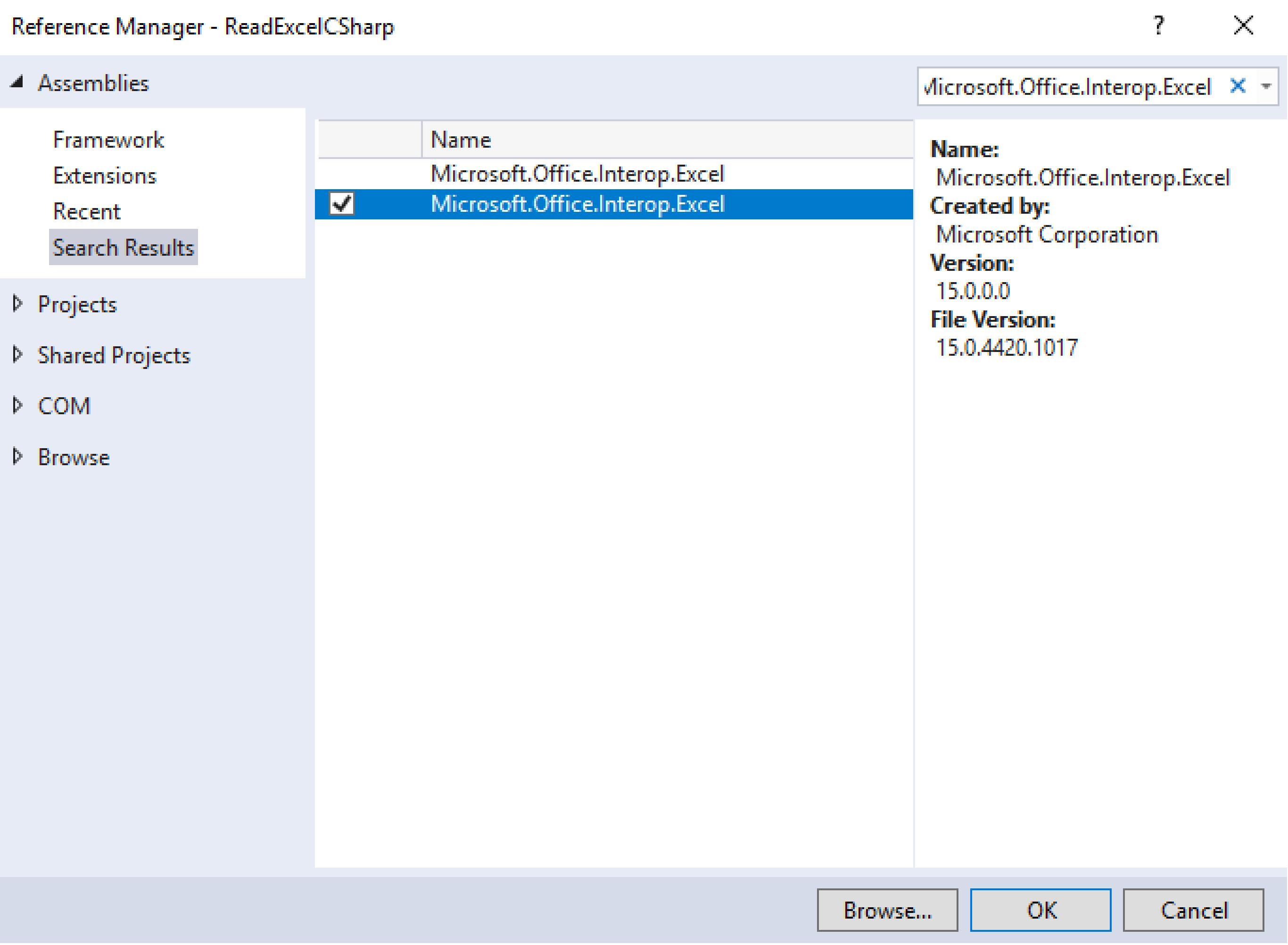
	A	B	C	D	E	F	G	H	I
1	0	First Name	Last Name	Gender	Country	Age	Date	Id	
2	1	Dulce	Abril	Female	United States	32	15/10/2017	1562	
3	2	Mara	Hashimoto	Female	Great Britain	25	16/08/2016	1582	
4	3	Philip	Gent	Male	France	36	21/05/2015	2587	
5	4	Kathleen	Hanner	Female	United States	25	15/10/2017	3549	
6	5	Nereida	Magwood	Female	United States	58	16/08/2016	2468	
7	6	Gaston	Brumm	Male	United States	24	21/05/2015	2554	
8	7	Etta	Hurn	Female	Great Britain	56	15/10/2017	3598	
9	8	Earlean	Melgar	Female	United States	27	16/08/2016	2456	
10	9	Vincenza	Weiland	Female	United States	40	21/05/2015	6548	

Sample Excel (.xlsx) file

Read Excel using MS Office Interop

You can also read Excel file in C# using MS Office Interop easily.

First you will have to add reference for " `Microsoft.Office.Interop.Excel` ", so in your Console, Windows or Web-Application, right-click on "Reference" and then in "Assemblies" search for "Microsoft.Office.Interop.Excel"



Microsoft.Office.Interop.Excel

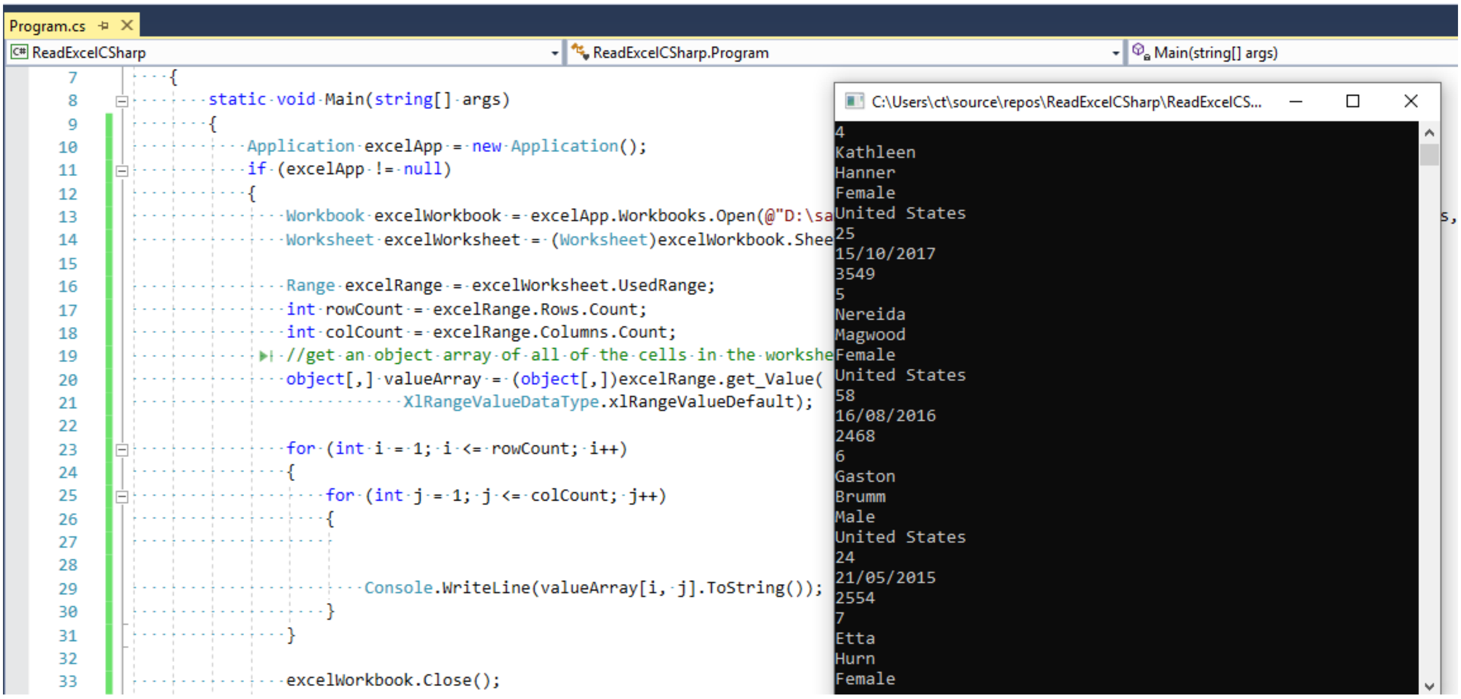
Microsoft.Office.Interop.Excel

☑ Microsoft.Office.Interop.Excel

Name: Microsoft.Office.Interop.Excel
Created by: Microsoft Corporation
Version: 15.0.0.0
File Version: 15.0.4420.1017

So, here is the C# Code for it, considering we have sample XLSX file as above

```
using Microsoft.Office.Interop.Excel;
using System;

namespace ReadExcelCSharp
{
    class Program
    {
        static void Main(string[] args)
        {
            Application excelApp = new Application();
            if (excelApp != null)
            {
                Workbook excelWorkbook = excelApp.Workbooks.Open(@"D:\sample_XLSX.xlsx", 0, true, 5, ""
, "", true, XlPlatform.xlWindows, "\t", false, false, 0, true, 1, 0);
                Worksheet excelWorksheet = (Worksheet)excelWorkbook.Sheets[1];

                Range excelRange = excelWorksheet.UsedRange;
                int rowCount = excelRange.Rows.Count;
                int colCount = excelRange.Columns.Count;
                //get an object array of all of the cells in the worksheet (their values)
                object[,] valueArray = (object[,])excelRange.get_Value(
                            XlRangeValueDataType.xlRangeValueDefault);

                for (int i = 1; i <= rowCount; i++)
                {
                    for (int j = 1; j <= colCount; j++)
                    {


                        Console.WriteLine(valueArray[i, j].ToString());
                    }
                }

                excelWorkbook.Close();
                excelApp.Quit();
            }
        }
    }
}
```



```
    {
        static void Main(string[] args)
        {
            Application excelApp = new Application();
            if (excelApp != null)
            {
                Workbook excelWorkbook = excelApp.Workbooks.Open(@"D:\sa
                Worksheet excelWorksheet = (Worksheet)excelWorkbook.Shee

                Range excelRange = excelWorksheet.UsedRange;
                int rowCount = excelRange.Rows.Count;
                int colCount = excelRange.Columns.Count;
                //get an object array of all of the cells in the workshe
                object[,] valueArray = (object[,])excelRange.get_Value(
                                XlRangeValueDataType.xlRangeValueDefault);

                for (int i = 1; i <= rowCount; i++)
                {
                    for (int j = 1; j <= colCount; j++)
                    {


                        Console.WriteLine(valueArray[i, j].ToString());
                    }
                }

                excelWorkbook.Close();
```

```
4
Kathleen
Hanner
Female
United States
25
15/10/2017
3549
5
Nereida
Magwood
Female
United States
58
16/08/2016
2468
6
Gaston
Brumm
Male
United States
24
21/05/2015
2554
7
Etta
Hurn
Female
```

As you can see we have used Interop to open and read excel rows/columns, one of the drawbacks of using MS office Interop is that you need it installed on Server also.

You may also like to read:

Read Excel file and import data into GridView using Datatable in ASP.NET (/article/asp-net/read-excel-file-and-import-data-into-gridview-using-datatabl/209)

Read file in C# (Text file example using Console Application) (/article/c-sharp/read-file-in-c-text-file-example-using-console-application/262)

That's it, feel free to provide your feedback in the below comment's section.

0

Comment

X

Comment's

(/users/profiles/278)

Steven P (/users/profiles/278)

Very useful and awesome post. I thank you for teaching concepts in deep. You can use ZetExcel.com for NET speeds up the spreadsheet processing and conversion tasks.

^ 1 ⌄ ⚠ Add Comment12/13/2019 5:38:03 AM

Login (/login) or Register (/register) to comment

X

Related Articles

👁2229	Using Generics in C# (With Example) (/article/c-/using-generics-in-c-with-example/103)
👁5732	Solving error "the breakpoint will not currently be hit" in Visual studio (Multiple ways) (/article/asp-net/solving-error-the-breakpoint-will-not-currently-be-hit-in-vi/257)

Subscribe Now

Subscribe to our weekly Newsletter & Keep getting latest article/questions in your inbox weekly

✉ your@email.com

SUBSCRIBE NOW!

X

(https://www.facebook.com/qawithexperts/)(https://twitter.com/qawithexperts)

Related Questions

👁12687	400. That’s an error. Error: redirect_uri_mismatch - Google OAuth Authentication (/questions/1/400-thats-an-error-error-redirecturimismatch)
👁6775	how to generate dynamic url using .NET MVC (/questions/4/how-to-generate-dynamic-url-using-net-mvc)
👁4489	How to convert JSON String into C# class object (/questions/7/how-to-convert-json-string-into-c-class-object)
👁10693	Cannot convert null to a value type JSON error (/questions/10/cannot-convert-null-to-a-value-type-json-error)
👁5985	DbArithmeticExpression arguments must have a numeric common type (/questions/11/dbarithmeticexpression-arguments-must-have-a-numeric-common)

Privacy Policy (/details/privacy-policy)

Minify HTML online (/tools/minify-html-online)
Minify CSS online (/tools/minify-css-online)
Minify JS (Javascript) online (/tools/minify-js-online)
Online HTML formatter (Beautifier) (/tools/html-beautifier-online)
Online CSS Beautifier (/tools/css-beautifier-online)
Online JS (Javascript) Formatter (/tools/javascript-beautifier-online)

Complete C# Tutorial (/tutorial/c-sharp)
Online XML Formatter (/tools/xml-formatter-online)

X

X

SIGN UP TO JOIN

(https://www.facebook.com/qawithexperts/)(https://twitter.com/qawithexperts)

X