

How to create an Excel file in .NET using OpenXML – Part 1: Basics

by Behnam (<http://www.dispatchertimer.com/author/admin/>) / 📅 16. November 2015 / 🏠 Tutorial
(<http://www.dispatchertimer.com/category/tutorial/>) / 💬 25

Overview

This is the 1st part of OpenXML Spreadsheet Tutorial Series:

- **Part 1: Basic**
- Part 2: Export a Collection to Spreadsheet (<http://www.dispatchertimer.com/tutorial/how-to-create-an-excel-file-in-net-using-openxml-part-2-export-a-collection-to-spreadsheet/>)
- Part 3: Add Stylesheet to the Spreadsheet (<http://www.dispatchertimer.com/tutorial/how-to-create-an-excel-file-in-net-using-openxml-part-3-add-stylesheet-to-the-spreadsheet/>)
- Part 4: Draw chart on Spreadsheet (<http://www.dispatchertimer.com/tutorial/how-to-create-an-excel-file-in-net-using-openxml-part-4-draw-chart/>)

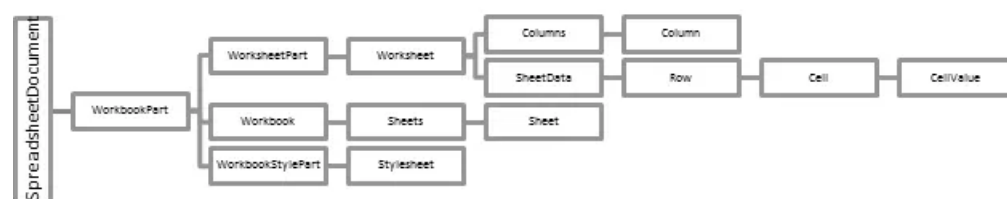
There are two ways which .NET helps you create an Excel file in .NET:

- Office Interop ([https://msdn.microsoft.com/en-us/library/microsoft.office.interop.excel\(v=office.15\).aspx](https://msdn.microsoft.com/en-us/library/microsoft.office.interop.excel(v=office.15).aspx))
- OpenXML 2.5 (<https://msdn.microsoft.com/en-us/library/office/bb448854.aspx>)

Office Interop needs Excel driver to be installed on the machine which is using it. In this tutorial we use OpenXML to be independent from having Excel driver.

With OpenXML you are simply constructing an XML structure which represents an Excel file. You start with creating a SpreadsheetDocument which represents an Excel document. Then you need to add workbook and worksheet parts in order to construct the document body.

The following diagram demonstrates the components in a Spreadsheet Document and their hierarchy:



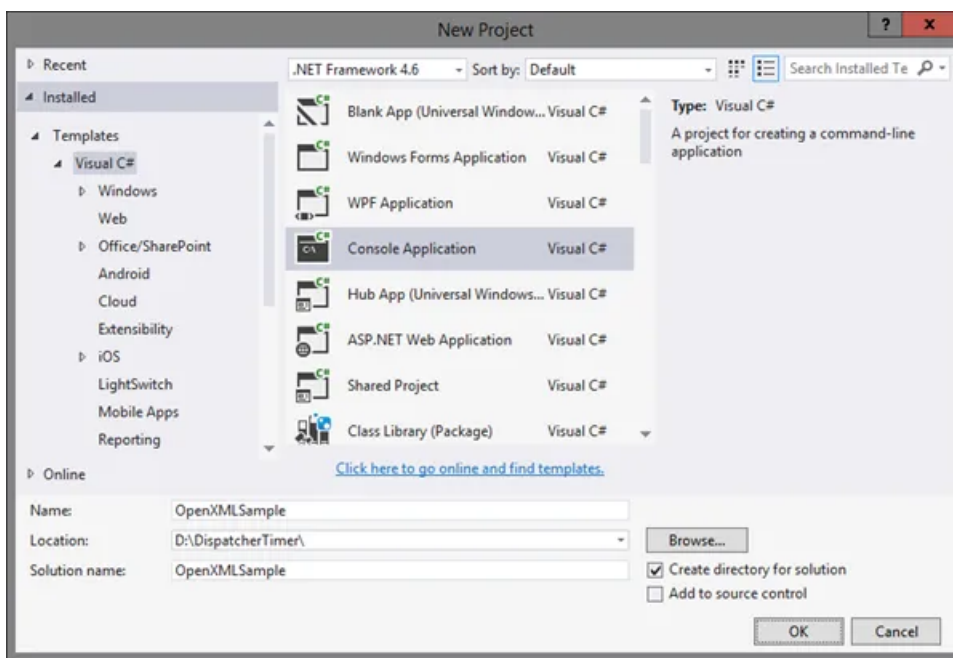
Don't worry if you don't understand this diagram at this moment. As we are going through the steps you will discover this hierarchy.

Sample: Constructing an Empty Excel File

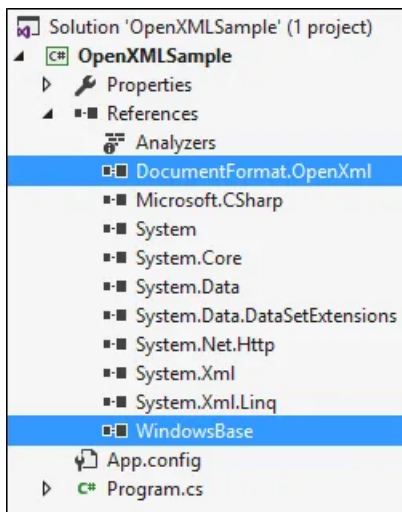
Open XML 2.5, C#, and .NET 4.6 with Visual Studio 2015 is used in this tutorial. You can still follow with older versions.

In the following sample we will only create an empty excel file without any actual data. It is important to understand the basics of how a document constructed.

- Open Visual Studio 2015 and create a new C# Console Application under: Installed, Templates, Visual C#



- You need to add reference to the following libraries:
 - DocumentFormat.OpenXml
 - WindowsBase



- Add a new class and name it "Report" and create a public method called "CreateExcelDoc". We are going to create and save our Excel file in this method.
- Import the following namespaces into the class:

```
using DocumentFormat.OpenXml;
using DocumentFormat.OpenXml.Packaging;
using DocumentFormat.OpenXml.Spreadsheet;
```

- Create a new SpreadsheetDocument and pass the file name and document as the parameters.

```
using (SpreadsheetDocument document = SpreadsheetDocument.Create(fileName, SpreadsheetDocumentType.Workbook))
{
}
```

- A SpreadsheetDocument must have at least a WorkbookPart and a WorkSheetPart.

Add the following code within the using block.

```
// Add a WorkbookPart to the document.
WorkbookPart workbookPart = document.AddWorkbookPart();
workbookPart.Workbook = new Workbook();
// Add a WorksheetPart to the WorkbookPart.
WorksheetPart worksheetPart = workbookPart.AddNewPart<WorksheetPart>();
worksheetPart.Worksheet = new Worksheet(new SheetData());
```

All the spreadsheet elements will be related via parent/child relationship.

The Workbook will contain our spreadsheet sheets.

The Worksheet will contain SheetData and Columns. It is the Sheet data which the actual values goes in rows and cells. By initializing the Worksheet we can append a SheetData as its child by passing it as argument.

- Append a "Sheets" to the Workbook. The Sheets will contain one or many Sheet which each associated with a WorksheetPart.

```
sheets = workbookPart.Workbook.AppendChild(new Sheets());
```

Then we can add one or many "Sheet" to the Sheets. These will be our Excelsheets. Take note that the sheet is associated to the WorksheetPart.

```
Sheet sheet = new Sheet() { Id = workbookPart.GetIdOfPart(worksheetPart), SheetId = 1, Name = "Test Sheet" };
sheets.Append(sheet);
```

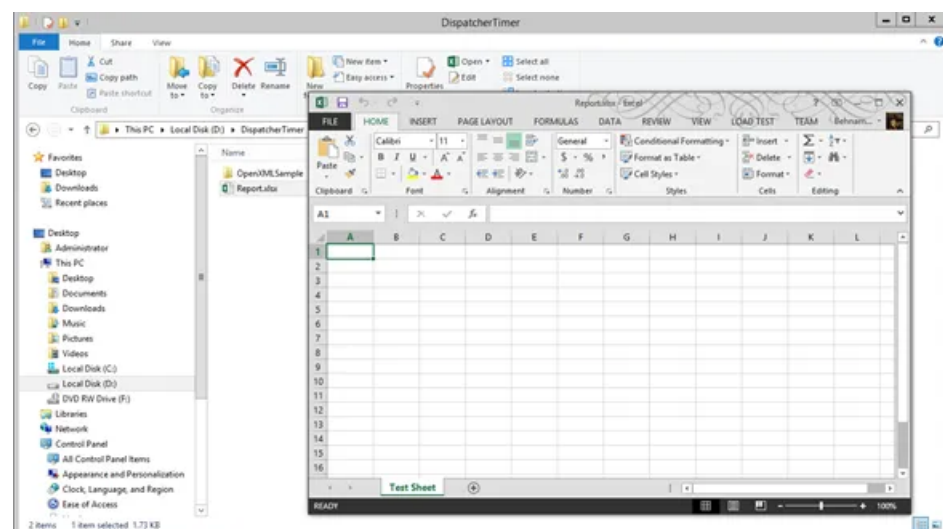
- At the end save the Workbook.

```
workbookPart.Workbook.Save();
```

- In the Main method of the Program class, create an object of the Report class and call the CreateExcelDocument by passing the file path.

```
static void Main(string[] args)
{
    Report report = new Report();
    report.CreateExcelDoc(@"D:\DispatcherTimer\Report.xlsx");
}
```

- Run the Project and check the generated Excel file:



Complete Code:

```

using DocumentFormat.OpenXml;
using DocumentFormat.OpenXml.Packaging;
using DocumentFormat.OpenXml.Spreadsheet;

namespace OpenXMLSample
{
    public class Report
    {
        public void CreateExcelDoc(string fileName)
        {
            using (SpreadsheetDocument document = SpreadsheetDocument.Create(fileName, SpreadsheetDocumentType.Workbook))
            {
                WorkbookPart workbookPart = document.AddWorkbookPart();
                workbookPart.Workbook = new Workbook();

                WorksheetPart worksheetPart = workbookPart.AddNewPart<WorksheetPart>();
                worksheetPart.Worksheet = new Worksheet(new SheetData());

                Sheets sheets = workbookPart.Workbook.AppendChild(new Sheets());

                Sheet sheet = new Sheet() { Id = workbookPart.GetIdOfPart(worksheetPart), SheetId = 1, Name = "Test" };

                sheets.Append(sheet);

                workbookPart.Workbook.Save();
            }
        }
    }
}

```


Farther reading


- Welcome to the Open XML SDK 2.5 for Office (<https://msdn.microsoft.com/en-us/library/office/bb448854.aspx>)
- Working with SpreadsheetML documents (Open XML SDK) (<https://msdn.microsoft.com/EN-US/library/office/gg278328.aspx>)

Download

Download the source code (<http://www.dispatchertimer.com/downloadcontents/OpenXMLSample-basics.zip>)

Share this:

 Print (<http://www.dispatchertimer.com/tutorial/how-to-create-an-excel-file-in-net-using-openxml-part-1-basics/#print>)

 Email (<http://www.dispatchertimer.com/tutorial/how-to-create-an-excel-file-in-net-using-openxml-part-1-basics/?share=email&nb=1>)

[Tweet](#)

 [Share](#)

Like this:

Loading...

Related

How to create an Excel file in .NET using OpenXML – Part 2: Export a collection to Spreadsheet
(<http://www.dispatchertimer.com/tutorial/how-to-create-an-excel-file-in-net-using-openxml-part-2-export-a-collection-to-spreadsheet/>)
November 19, 2015
In "Tutorial"

Troubleshooting OpenXML Spreadsheet
(<http://www.dispatchertimer.com/troubleshooting-openxml-spreadsheet/>)
November 24, 2015
In "Troubleshooting"

How to create an Excel file in .NET using OpenXML – Part 3: Add Stylesheet to the Spreadsheet
(<http://www.dispatchertimer.com/tutorial/how-to-create-an-excel-file-in-net-using-openxml-part-3-add-stylesheet-to-the-spreadsheet/>)
November 20, 2015
In "Tutorial"

25

thoughts on “How to create an Excel file in .NET using OpenXML – Part 1: Basics”



1

Don on February 7, 2017

Thank you so much for your effort and time to share your knowledge on OpenXML and creating a spreadsheet document. This overview and tutorial has saved me much time and frustration. This is very much appreciated.



Behnam on February 8, 2017

Hi Don. You are welcome. I am happy it was helpful.



Ana on June 15, 2017

Nice tutorial, helped me a lot.

One question, how can I create and open next the file? Do you know?

Thanks



Behnam on June 19, 2017

You are welcome.

These tutorials are focused on writing into Excel. In order to read you can use the following snippet to start with.:

```
var fileStream = new FileStream(fileName, FileMode.Open, FileAccess.Read, FileShare.ReadWrite);  
var spreadsheetDocument = SpreadsheetDocument.Open(fileStream, false);  
WorkbookPart workbookPart = spreadsheetDocument.WorkbookPart;  
WorksheetPart worksheetPart = workbookPart.WorksheetParts.First();  
Worksheet sheetPart = worksheetPart.Worksheet;  
List rows = sheetPart.Descendants().ToList();
```

```
foreach (var row in rows )  
{  
    // do whatever you want with the row  
}
```



5
Carlos F Vargas on July 19, 2017

Saludos desde Colombia, Muchas gracias por el tutorial.



Behnam on July 19, 2017

De nada.



7
Basha Himmatgori on August 7, 2017

Thanks Behnam. I was struggling to kick start my openXML journey. your example helped a lot.



Behnam on August 8, 2017

Hi Basha. You are welcome. I am happy the article helps you.



9
Cahit on September 19, 2017

Thank you for tutorial.

I ran the code it worked but excel file didnt open. There were no error on running. Do you know why?



Behnam on September 21, 2017

Hi. The part 1 of the tutorial series only creates an empty sheet. It is just to teach the basic architecture and an Excel sheet. If it opens blank that's correct. If it gives some error, let me know that.



11
Jimmy (<http://www.marindataanalytics.com>) on February 15, 2018

I don't understand what this filename is. Whenever I enter it says access it denied to the file. Why do we even need a file name? I thought this was only going to create and open up a blank worksheet.



Behnam on April 6, 2018

You are right, this section only demonstrates how to create an empty sheet and save it to a file on disc. Hence it needs a file name. Why it gives you an access denied message is where you are pointing your program to creates the file. You might check if necessary accesses have given to that folder.



Manu on April 12, 2018

Will this work on Xamarin (Android and IOS)?
If yes, what type is List in your code example?:
"List rows = sheetPart.Descendants().ToList();"

Visual Studio doesn't recognize it. Do you mean List of something?

Thanks



Zach on July 10, 2018

Haven't gotten this to work quite yet but OpenXmlType is the data type referenced.

"List rows = sheetPart.Descendants().ToList();"

Buyer beware, I am an idiot.



Zach on July 10, 2018

It's an OpenXmlElement. Had a mashup of different tutorials going on. Sure wish I could delete or edit my reply.



Martha Ng on April 18, 2018

Hola,

Muy interesante, estoy intentando hacer un XML y este pasarlo a Excel, pasándole un dataset con varias tablas de sql, tengo la duda de ¿cómo le podría hacer para que cada tabla fuese una hoja en el XML ? y así ver cada tabla en una hoja al abrirlo en el Excel



17

Geetha on June 19, 2018

If we want to protect the spreadsheet with password, what we need to do?



18

mayura on August 6, 2018

I want to create cell with checkbox in excel using openxml. Does anyone has idea about this.



19

sanjaytyata on December 24, 2018

Need help for creating dynamic excel ie; dropdown in multiple columns with datavalidation



20

Jkke Sparrow on January 11, 2019

How do we prevent our computers region settings from overriding the date format specified in c# ?



21

Robert on June 26, 2019

Hi Behnam,

first of all thanks for your effort. That was grate for as OpenXML starter.

I have two questions:

1. Why you have not defined any RowIndex for your rows? This is also the case with the cells created with "ConstructCell" as there is no definition for "CellReference" there. If the rows and cells are created so, then there is no chance to refere them later.

My sugesstion is:

for the rows:

```
private Row CreateRow(uint rowIndex)
{
    return new Row() { RowIndex = rowIndex };
}
```

and the for cells at least "CellReference" exists in the your "ConstructCell" parameter list.

2. I have replaced your construction for columns with the folowing. But my column definitions cannot be seen in the result.

```
var columns = new Columns();
var column1 = new Column { Min = 1, Max = 1, Width = 4, CustomWidth = true };
var column2 = new Column { Min = 1, Max = 1, Width = 20, CustomWidth = true };
var column3 = new Column { Min = 1, Max = 1, Width = 30, CustomWidth = true };
var column4 = new Column { Min = 1, Max = 1, Width = 10, CustomWidth = true };
```

```
columns.Append(column1);
columns.Append(column2);
columns.Append(column3);
```



```
columns.Append(column4);  
  
//worksheetPart.Worksheet.Append(columns);  
  
worksheetPart.Worksheet.AppendChild(columns);  
Kind regards
```

 *Sandeep Singh* on July 12, 2019

How to create line chart?

 *Pep* on September 12, 2019

Thanks, for large file production what do you suggest ?

 *Reader* on November 19, 2019

A mistake on the image, that the WorksheetPart should be WorksheetParts -> WorksheetPart. Each sheet relates to one WorksheetPart, and each WorksheetPart has one Worksheet, each Worksheet has one SheetData.

 *Brijen Makwana* on May 19, 2020

Thank you for this tutorial. It was quite helpful.

One more thing I want to ask that Can we have Import and Export Excel file in OpnXML with configurable part? Like suppose I am importing one excel with some styles and data and I try to read the data from it and apply some calculations on it.

Now suppose one day someone just update the excel template and add one new row in it. So without changing the code I just need to update the configuration file and change the cell names (suppose from C5 to C6) and now calculations should be done on the C6 value.

Could you please throw some light on it if you have ever come across some kind of situation?

Thanks in advance.



Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

☐

Save my name, email, and website in this browser for the next time I comment.

☐

Notify me of follow-up comments by email.

☐

Notify me of new posts by email.

POST COMMENT

This site uses Akismet to reduce spam. [Learn how your comment data is processed \(https://akismet.com/privacy/\)](https://akismet.com/privacy/).

[Previous Pagination with Filtering using AngularJS \(http://www.dispatchertimer.com/angularjs/pagination-with-filtering-using-angularjs/\)](http://www.dispatchertimer.com/angularjs/pagination-with-filtering-using-angularjs/)

[Next How to create an Excel file in .NET using OpenXML – Part 2: Export a collection to Spreadsheet \(http://www.dispatchertimer.com/tutorial/how-to-create-an-excel-file-in-net-using-openxml-part-2-export-a-collection-to-spreadsheet/\)](http://www.dispatchertimer.com/tutorial/how-to-create-an-excel-file-in-net-using-openxml-part-2-export-a-collection-to-spreadsheet/)

Search ...

Q

CATEGORIES

- > [ANGULARJS \(HTTP://WWW.DISPATCHERTIMER.COM/CATEGORY/ANGULARJS/\)](http://www.dispatchertimer.com/category/angularjs/)
- > [POWERSHELL \(HTTP://WWW.DISPATCHERTIMER.COM/CATEGORY/POWERSHELL/\)](http://www.dispatchertimer.com/category/powershell/)
- > [SCRUM \(HTTP://WWW.DISPATCHERTIMER.COM/CATEGORY/SCRUM/\)](http://www.dispatchertimer.com/category/scrum/)
- > [SIGNALR \(HTTP://WWW.DISPATCHERTIMER.COM/CATEGORY/SIGNALR/\)](http://www.dispatchertimer.com/category/signalr/)
- > [SQL SERVER \(HTTP://WWW.DISPATCHERTIMER.COM/CATEGORY/SQL-SERVER/\)](http://www.dispatchertimer.com/category/sql-server/)
- > [TROUBLESHOOTING \(HTTP://WWW.DISPATCHERTIMER.COM/CATEGORY/TROUBLESHOOTING/\)](http://www.dispatchertimer.com/category/troubleshooting/)

- > TUTORIAL ([HTTP://WWW.DISPATCHERTIMER.COM/CATEGORY/TUTORIAL/](http://www.dispatchertimer.com/category/tutorial/))
- > WCF ([HTTP://WWW.DISPATCHERTIMER.COM/CATEGORY/WCF/](http://www.dispatchertimer.com/category/wcf/))
- > WEB API ([HTTP://WWW.DISPATCHERTIMER.COM/CATEGORY/WEB-API/](http://www.dispatchertimer.com/category/web-api/))
- > WORKFLOW FOUNDATION ([HTTP://WWW.DISPATCHERTIMER.COM/CATEGORY/WORKFLOW-FOUNDATION/](http://www.dispatchertimer.com/category/workflow-foundation/))

SUBSCRIBE TO DISPATCHERTIMER

Enter your email address to subscribe to this blog and receive notifications of new posts by email.

TAGS

[AGILE \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/AGILE/\)](http://www.dispatchertimer.com/tag/agile/) [ANGULARJS \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/ANGULARJS/\)](http://www.dispatchertimer.com/tag/angularjs/) [DUPLEX COMMUNICATION \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/DUPLEX-COMMUNICATION/\)](http://www.dispatchertimer.com/tag/duplex-communication/) [EXCEL \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/EXCEL/\)](http://www.dispatchertimer.com/tag/excel/) [EXCEPTION HANDLING \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/EXCEPTION-HANDLING/\)](http://www.dispatchertimer.com/tag/exception-handling/) [OFFICE \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/OFFICE/\)](http://www.dispatchertimer.com/tag/office/) [OPENXML \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/OPENXML/\)](http://www.dispatchertimer.com/tag/openxml/) [POWERSHELL \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/POWERSHELL/\)](http://www.dispatchertimer.com/tag/powershell/) [SCRUM \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/SCRUM/\)](http://www.dispatchertimer.com/tag/scrum/) [SIGNALR \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/SIGNALR/\)](http://www.dispatchertimer.com/tag/signalr/) [SOFTWARE DEVELOPMENT METHODOLOGY \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/SOFTWARE-DEVELOPMENT-METHODOLOGY/\)](http://www.dispatchertimer.com/tag/software-development-methodology/) [SQLSERVER \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/SQLSERVER/\)](http://www.dispatchertimer.com/tag/sqlserver/) **WCF** ([HTTP://WWW.DISPATCHERTIMER.COM/TAG/WCF/](http://www.dispatchertimer.com/tag/wcf/)) [WCF CONFIGURATION \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/WCF-CONFIGURATION/\)](http://www.dispatchertimer.com/tag/wcf-configuration/) [WEB API \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/WEB-API/\)](http://www.dispatchertimer.com/tag/web-api/) [WF \(HTTP://WWW.DISPATCHERTIMER.COM/TAG/WF/\)](http://www.dispatchertimer.com/tag/wf/)

RECENT POSTS

- > [BEST PRACTICE IN CREATING WEB API SERVICES \(HTTP://WWW.DISPATCHERTIMER.COM/WEB-API/BEST-PRACTICE-IN-CREATING-WEB-API-SERVICES/\)](http://www.dispatchertimer.com/web-api/best-practice-in-creating-web-api-services/)
- > [WEB API TUTORIAL: HOW TO CREATE A SIMPLE HTTP SERVICE USING WEB API \(HTTP://WWW.DISPATCHERTIMER.COM/WEB-API/WEB-API-TUTORIAL-HOW-TO-CREATE-A-SIMPLE-HTTP-SERVICE-USING-WEB-API/\)](http://www.dispatchertimer.com/web-api/web-api-tutorial-how-to-create-a-simple-http-service-using-web-api/)
- > [HOW TO DEFINE CLASSES IN POWERSHELL \(HTTP://WWW.DISPATCHERTIMER.COM/POWERSHELL/HOW-TO-DEFINE-CLASSES-IN-POWERSHELL/\)](http://www.dispatchertimer.com/powershell/how-to-define-classes-in-powershell/)
- > [HOW TO CREATE AN EXCEL FILE IN .NET USING OPENXML – PART 4: DRAW CHART \(HTTP://WWW.DISPATCHERTIMER.COM/TUTORIAL/HOW-TO-CREATE-AN-EXCEL-FILE-IN-NET-USING-OPENXML-PART-4-DRAW-CHART/\)](http://www.dispatchertimer.com/tutorial/how-to-create-an-excel-file-in-net-using-openxml-part-4-draw-chart/)
- > [TROUBLESHOOTING OPENXML SPREADSHEET \(HTTP://WWW.DISPATCHERTIMER.COM/TROUBLESHOOTING/TROUBLESHOOTING-OPENXML-SPREADSHEET/\)](http://www.dispatchertimer.com/troubleshooting/troubleshooting-openxml-spreadsheet/)

Threading.DispatcherTimer.com

 (<https://twitter.com/bendiso>)  (<https://plus.google.com/+DispatchertimerTips>)

 (<https://my.linkedin.com/in/behnamkarimi>)



Home (<http://www.dispatchertimer.com>) - WCF Tutorial (<http://www.dispatchertimer.com/wcf/windows-communication-foundation/>) -
WF Tutorial (<http://www.dispatchertimer.com/tutorial/everything-about-windows-workflow-foundation-wf/>) -
Windows PowerShell (<http://www.dispatchertimer.com/tag/powershell/>) - About Me (<http://www.dispatchertimer.com/about-me/>)