

Excel for .NET

Copyright © 1987-2013 GrapeCity, Inc. All rights reserved.

ComponentOne, a division of GrapeCity

201 South Highland Avenue, Third Floor

Pittsburgh, PA 15206 • USA

Internet: info@ComponentOne.com

Web site: <http://www.componentone.com>

Sales

E-mail: sales@componentone.com

Telephone: 1.800.858.2739 or 1.412.681.4343 (Pittsburgh, PA USA Office)

Trademarks

The ComponentOne product name is a trademark and ComponentOne is a registered trademark of GrapeCity, Inc. All other trademarks used herein are the properties of their respective owners.

Warranty

ComponentOne warrants that the original CD (or diskettes) are free from defects in material and workmanship, assuming normal use, for a period of 90 days from the date of purchase. If a defect occurs during this time, you may return the defective CD (or disk) to ComponentOne, along with a dated proof of purchase, and ComponentOne will replace it at no charge. After 90 days, you can obtain a replacement for a defective CD (or disk) by sending it and a check for \$25 (to cover postage and handling) to ComponentOne.

Except for the express warranty of the original CD (or disks) set forth here, ComponentOne makes no other warranties, express or implied. Every attempt has been made to ensure that the information contained in this manual is correct as of the time it was written. We are not responsible for any errors or omissions. ComponentOne's liability is limited to the amount you paid for the product. ComponentOne is not liable for any special, consequential, or other damages for any reason.

Copying and Distribution

While you are welcome to make backup copies of the software for your own use and protection, you are not permitted to make copies for the use of anyone else. We put a lot of time and effort into creating this product, and we appreciate your support in seeing that it is used by licensed users only.

This manual was produced using [ComponentOne Doc-To-Help™](#).

Table of Contents

ComponentOne Excel for .NET Overview	3
Help with ComponentOne Studio for WinForms.....	3
Migrating an Excel for .NET Project to Visual Studio 2005	3
Excel for .NET Limitations	4
Key Features.....	5
Excel for .NET QuickStart	6
Step 1 of 4: Setting up the Project.....	7
Step 2 of 4: Adding Content to a C1XLBook.....	7
Step 3 of 4: Formatting the Content.....	8
Step 4 of 4: Saving and Opening the XLS File.....	10
Using Excel for .NET	11
Creating Documents	12
Worksheets	15
Rows and Columns.....	16
Cells.....	16
Styles.....	17
Excel for .NET Frequently Asked Questions.....	17
Excel for .NET Task-Based Help.....	18
Adding Content to a Workbook.....	19
Merging Cells	21
Formatting Cells	22
Copying Rows from One Book to Another.....	24
Adding an Image to a Cell	26
Adding a Comment to a Cell	33
Adding a Page Break to a Worksheet	35
Setting the Calculation Mode for a Workbook.....	36
Importing and Exporting OpenXml Files.....	38
Creating Subtotals.....	40
Saving and Loading CSV Files	43

Excel for .NET Samples.....	47
API Reference.....	49
C1.C1Excel.4 Assembly	49
Overview	49
Namespaces	50
C1.C1Excel Namespace	50
C1.Win.Localization Namespace.....	543

ComponentOne Excel for .NET Overview

Your Excel data is just a simple command away from any of your .NET applications with **ComponentOne Excel for .NET** you don't even need to have Microsoft Excel installed! Create or load XLS files for Excel 97 and later. **Excel for .NET** supports the new Office 2007 OpenXml format, which allows you to save smaller, compressed XLSX files.

The main component in **Excel for .NET** is the C1XLBook object, which represents an Excel workbook containing one or more sheets. Use the C1XLBook to load existing Excel files or create new ones. Then add sheets, styles, hyperlinks, images, headers and footers, page breaks and more. When you are done, save the C1XLBook to a file or a Stream and you're done. Anyone with a copy of Excel can access your data. It's that easy!

See Also

[Help with ComponentOne Studio for WinForms](#)

[Migrating an Excel for .NET Project to Visual Studio 2005](#)

[Excel for .NET Limitations](#)

Help with ComponentOne Studio for WinForms

Getting Started

For information on installing ComponentOne Studio for WinForms, licensing, technical support, namespaces and creating a project with the control, please visit [Getting Started with Studio for WinForms](#).

What's New

For a list of the latest features added to **ComponentOne Studio for WinForms**, visit [What's New in Studio for WinForms](#).

Migrating an Excel for .NET Project to Visual Studio 2005

To migrate a project using ComponentOne components to Visual Studio 2005, there are two main steps that must be performed. First, you must convert your project to Visual Studio 2005, which includes removing any references to a previous assembly and adding a reference to the new assembly. Secondly, the .licx file, or licensing file, must be updated in order for the project to run correctly.

To convert the project:

1. Open Visual Studio 2005 and select **File, Open Project**.

2. Locate the **.sln** file for the project that you wish to convert to Visual Studio 2005. Select it and click **Open**. The **Visual Studio Conversion Wizard** appears.
3. Click **Next**.
4. Select **Yes, create a backup before converting** to create a backup of your current project and click **Next**.
5. Click **Finish** to convert your project to Visual Studio 2005. The **Conversion Complete** window appears.
6. Click **Show the conversion log when the wizard is closed** if you want to view the conversion log.
7. Click **Close**. The project opens. Now you must remove references to any of the previous ComponentOne .dlls and add references to the new ones.
8. Go to the Solution Explorer (**View | Solution Explorer**), select the project, and click the **Show All Files** button.



Note: The **Show All Files** button does not appear in the Solution Explorer toolbar if the Solution project node is selected.

9. Expand the **References** node, right-click C1.Common and select **Remove**. Also remove c1.c1excel the same way.
10. Right-click the **References** node and select **Add Reference**.
11. Browse and select **C1.C1Excel.2.dll**. Click **OK** to add it to the project.

To update the .licx file:

1. 1. In the Solution Explorer, right-click the **licenses.licx** file and select **Delete**.
2. 2. Click **OK** to permanently delete **licenses.licx**. The project must be rebuilt to create a new, updated version of the .licx file.
3. 3. Click the **Start Debugging** button to compile and run the project. The new .licx file may not be visible in the Solution Explorer.
4. 4. Select **File, Close** to close the form and then double-click the **Form.vb** or **Form.cs** file in the Solution Explorer to reopen it. The new **licenses.licx** file appears in the list of files.

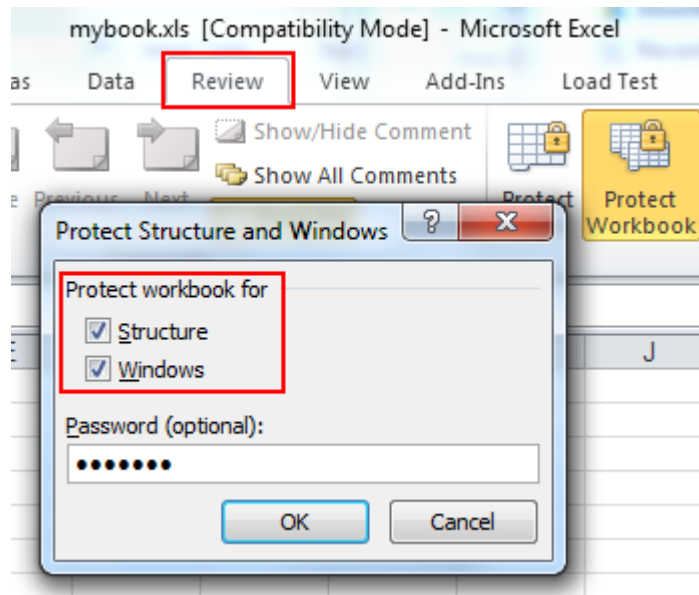
The migration process is complete.

Excel for .NET Limitations

- There is no support for the PivotTable declaration.
- There is no support for macros.
- When saving a formatted cell using the **Fill** property from an .xlsx to an .xls file, an ArgumentException is thrown if the rows and columns in the .xlsx format are more than the maximum for the .xls format.
- **C1Excel** does not support RTF (with bold/italic/underline text) records.
- A chart object is not saved when a BIFF8 file is loaded and saved using **C1Excel**.
- There is no support for the Protect Structure/Windows:

When **ComponentOne Excel for .NET** is used to load a Microsoft Excel file that is password protected for Structure/Windows: a "Central dir not found" exception or `FileLoadException` may occur.

The following image shows how a workbook has been password protected for Structure and Windows in Microsoft Excel 2010:



Key Features

The following are some of the main features of **ComponentOne Excel for .NET** that you may find useful:

- Save or load a workbook with one command

Excel for .NET is easy-to-use, allowing you to use a single command to load or save a workbook and manipulate sheets as if they were grid controls.

- Read and write data in individual cells

After loading or creating a [C1XLBook](#), you can access data in individual sheets as if they were a simple grid. For example:

```
XLSheet sheet = C1XLBook.Sheets[0];  
sheet[0, 0].Value = DateTime.Now;
```

- Format the data in each cell.

The format associated with each cell is as easy to access as the data stored in the cell. For example:

```
XLStyle style = new XLStyle(c1XLBook1);
style.Format = "dd-MM-yyyy";
style.Font = new Font("Courier New", 14);
XLSheet sheet = C1XLBook.Sheets[0];
sheet[0, 0].Value = DateTime.Now;
sheet[0, 0].Style = style;
```

- Use **Excel for .NET** to export to XLS files

Other ComponentOne components use **Excel for .NET** to export XLS files. For example, **C1Report** uses **Excel for .NET** to create XLS versions of reports so they can be viewed and edited by anyone with a copy of Microsoft Excel.

- Reads and writes .xls and .xlsx files without using Microsoft Excel

Excel for .NET reads and writes .xls (Excel 97 and later) and .xlsx (OpenXml format) files, the latter of which can be reused and easily exchanged or compressed to create smaller file sizes. You don't even need to have Microsoft Excel installed.

- Create and position images within a cell

Not only can you add images to cells, but now you can specify the cell size, the position of the image within the cell and whether the image is scaled, clipped, or stretched to fit the cell.

- Save and load files to and from streams

Workbooks can now be directly read to and written from memory streams with new overloads for the [Load](#) and the [Save](#) methods so you no longer have to use temporary files.

- Add images to the header and footer of a sheet

Use properties in the [XLPrintSettings](#) class to add images to the left, center, or right part of a sheet's header or footer.

Excel for .NET QuickStart

This quick start guide will familiarize you with some of the features of **Excel for .NET**. In this quick start you will learn how to add a C1XLBook to the project, add formatted data to the workbook, and save and open the XLS file.

See Also

[Step 1 of 4: Setting up the Project](#)

[Step 2 of 4: Adding Content to a C1XLBook](#)

[Step 3 of 4: Formatting the Content](#)

[Step 4 of 4: Saving and Opening the XLS File](#)

Step 1 of 4: Setting up the Project

In this step you will add a C1XLBook component to your form. Each book is composed of one or more sheets.

1. Create a new .NET 2.0 project.
2. In the Toolbox, double-click the **C1XLBook** icon to add the C1XLBook component to your project. The C1XLBook component will appear in the component tray below the form.
3. Double-click the form to add the **Form1_Load** event and switch to code view.
4. Add the **Imports** (Visual Basic) or **using** (C#) statement to the code at the top of the form so you can use all names within the C1.C1Excel namespace.

[To write code in Visual Basic](#)

Visual Basic	Copy Code
Imports C1.C1Excel	

[To write code in C#](#)

C#	Copy Code
using C1.C1Excel;	

Now that you have a C1XLBook, you can begin adding content to it.

Step 2 of 4: Adding Content to a C1XLBook

While you are still in code view in the Visual Studio project, add the following code within the **Form_Load** event created in [Step 1 of 4](#). This code will add content to the Excel workbook.

[To write code in Visual Basic](#)

Visual Basic	Copy Code
' Add content to the sheet.	

```

Dim i As Integer
Dim sheet as XLSheet = C1XLBook1.Sheets(0)
For i = 0 To 9
    sheet(i, 0).Value = (i + 1) * 10
    sheet(i, 1).Value = (i + 1) * 100
    sheet(i, 2).Value = (i + 1) * 1000
Next i

```

To write code in C#

C#	Copy Code
<pre> // Add content to the sheet. int i; C1.C1Excel.XLSheet sheet = c1XLBook1.Sheets[0]; for (i = 0; i <= 9; i++) { sheet[i, 0].Value = (i + 1) * 10; sheet[i, 1].Value = (i + 1) * 100; sheet[i, 2].Value = (i + 1) * 1000; } </pre>	

The first ten rows in the first three columns of the XLS file will be populated with numbers when you run the project.

Step 3 of 4: Formatting the Content

Next we will format the content using styles. The code in this step should be added after the code from [Step 2 of 4](#) within the **Form_Load** event.

1. Add the following code to create two new styles: style1 and style2.

To write code in Visual Basic

Visual Basic	Copy Code
<pre> 'Add style 1. Dim style1 As New XLStyle(C1XLBook1) </pre>	

```

style1.Font = New Font("Tahoma", 9, FontStyle.Bold)
style1.ForeColor = Color.RoyalBlue
' Add style 2.
Dim style2 As New XLStyle(C1XLBook1)
style2.Font = New Font("Tahoma", 9, FontStyle.Italic)
style2.BackColor = Color.RoyalBlue
style2.ForeColor = Color.White

```

To write code in C#

Title Text	Copy Code
<pre> // Add style 1. XLStyle style1 = new XLStyle(c1XLBook1); style1.Font = new Font("Tahoma", 9, FontStyle.Bold); style1.ForeColor = Color.RoyalBlue; // Add style 2. XLStyle style2 = new XLStyle(c1XLBook1); style2.Font = new Font("Tahoma", 9, FontStyle.Italic); style2.BackColor = Color.RoyalBlue; style2.ForeColor = Color.White; </pre>	

2. Then add the following code to apply the new styles to the content.

To write code in Visual Basic

Visual Basic	Copy Code
<pre> For i = 0 To 9 ' Apply styles to the content. If (i + 1) Mod 2 = 0 Then sheet(i, 0).Style = style2 sheet(i, 1).Style = style1 sheet(i, 2).Style = style2 Else </pre>	

```

        sheet(i, 0).Style = style1
        sheet(i, 1).Style = style2
        sheet(i, 2).Style = style1
    End If
Next i

```

To write code in C#

C#	Copy Code
<pre> for (i = 0; i <= 9; i++) { // Apply styles to the content. if ((i + 1) % 2 == 0) { sheet[i, 0].Style = style2; sheet[i, 1].Style = style1; sheet[i, 2].Style = style2; } else { sheet[i, 0].Style = style1; sheet[i, 1].Style = style2; sheet[i, 2].Style = style1; } } </pre>	

Step 4 of 4: Saving and Opening the XLS File

Finally, add the following code to save and load the Excel workbook. This code should be added after the code from [Step 3 of 4](#) within the **Form_Load** event.

To write code in Visual Basic

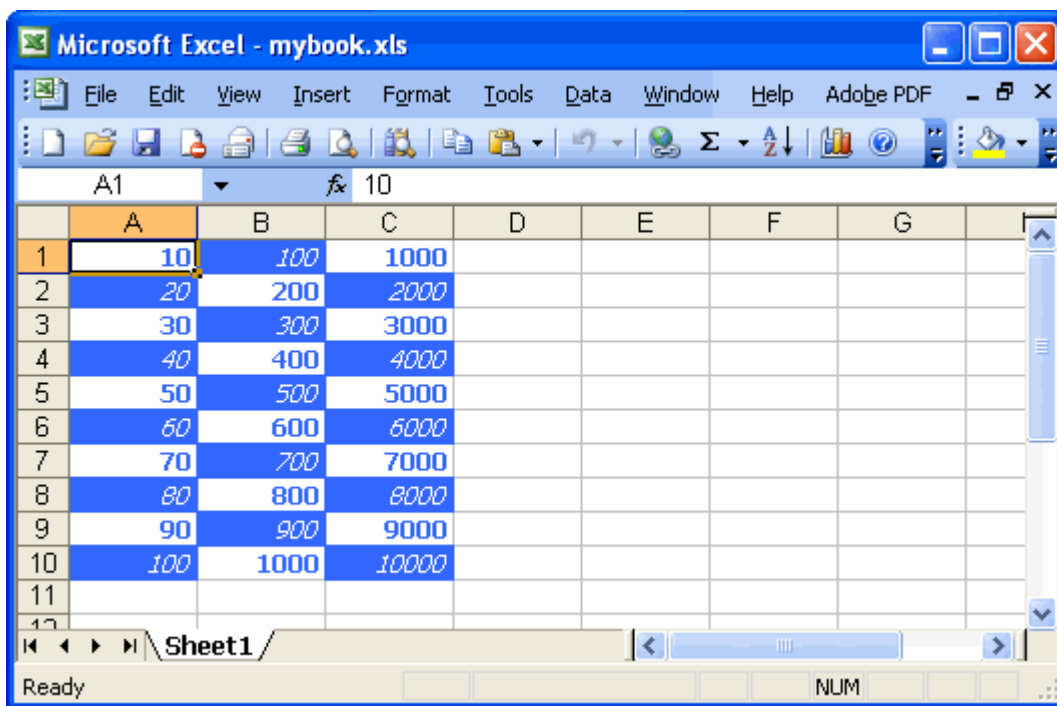
Visual Basic	Copy Code
<pre> C1XLBook1.Save("c:\mybook.xls") </pre>	

```
System.Diagnostics.Process.Start("C:\mybook.xls")
```

To write code in C#

C#	Copy Code
<pre>c1XLBook1.Save(@"c:\mybook.xls"); System.Diagnostics.Process.Start(@"c:\mybook.xls");</pre>	

Run the program and observe:



Formatted content is added to the workbook.

Congratulations! You've completed the **Excel for .NET** quick start.

Using Excel for .NET

The following topics explain how to create an XLS file, as well as describe the main **Excel for .NET** classes used to create the components that make up the file, which include worksheets, rows and columns, cells and styles.

See Also

[Creating Documents](#)

[Worksheets](#)

[Rows and Columns](#)

[Cells](#)

[Styles](#)

Creating Documents

To create a new XLS file using **Excel for .NET**, three steps are required:

1. Add a [C1XLBook](#) component to your project or create a C1XLBook object in code. Each book is composed of one or more sheets ([XLSheet](#) objects).
2. Add content to the sheets. Each sheet contains cells ([XLCell](#) objects) that have a [Value](#) and a [Style](#) property.
3. Save the book to a file using the [Save](#) method.

For example, the following code creates a new Excel file with a single sheet containing numbers from 1 to 100. Note that if you add the C1XLBook component to a form, you can skip the code in step 1.

[To write code in Visual Basic](#)

Visual Basic	Copy Code
<pre>' step 1: create a new workbook Dim C1XLBook1 As New C1XLBook() ' step 2: write content into some cells Dim sheet As XLSheet = C1XLBook1.Sheets(0) Dim i As Integer For i = 0 To 99 sheet(i, 0).Value = i + 1 Next i ' step 3: save the file C1XLBook1.Save("c:\temp\hello.xls")</pre>	

[To write code in C#](#)

C#	Copy Code
<pre>// step 1: create a new workbook C1XLBook c1XLBook1 = new C1XLBook(); // step 2: write content into some cells XLSheet sheet = c1XLBook1.Sheets[0]; for (int i = 0; i < 100; i++) { sheet[i, 0].Value = i + 1; } // step 3: save the file c1XLBook1.Save(@"c:\temp\hello.xls");</pre>	

Step 2 is the most interesting one. The code starts by retrieving an **XLSheet** object that represents the single worksheet in the new Excel workbook. This sheet is created automatically when you add or create a new **C1XLBook**. Then the code uses the sheet indexer to reference cells in the sheet and assign them values from 1 to 100.

Note that the indexer in the **XLSheet** object automatically creates cells, if necessary. This makes it easy to fill worksheets that you create. If you want to find out the sheet dimensions, use the sheet's **Rows.Count** and **Columns.Count** properties.

Of course, you are not limited to assigning values to cells. You can also use styles to format the cells. Just create one or more **XLStyle** objects and assign them to cells much like you did values. This revised version of the code above creates a sheet where even numbers are shown in bold red characters and odd numbers in italic blue. Note that if you add the **C1XLBook** component to the form, you can skip the code in step 1.

[To write code in Visual Basic](#)

Visual Basic	Copy Code
<pre>' step 1: create a new workbook Dim C1XLBook1 As New C1XLBook() ' step 2: create styles for odd and even values Dim styleOdd As New XLStyle(C1XLBook1) styleOdd.Font = New Font("Tahoma", 9, FontStyle.Italic) styleOdd.ForeColor = Color.Blue Dim styleEven As New XLStyle(C1XLBook1)</pre>	

```

styleEven.Font = New Font("Tahoma", 9, FontStyle.Bold)
styleEven.ForeColor = Color.Red
' step 3: write content and styles into some cells
Dim sheet As XLSheet = C1XLBook1.Sheets(0)
Dim i As Integer
For i = 0 To 99
    Dim cell As XLCell = sheet(i, 0)
    cell.Value = i + 1
    If (i + 1) Mod 2 = 0 Then
        cell.Style = styleEven
    Else
        cell.Style = styleOdd
    EndIf
Next i
' step 4: save the file
C1XLBook1.Save("c:\temp\hello.xls")

```

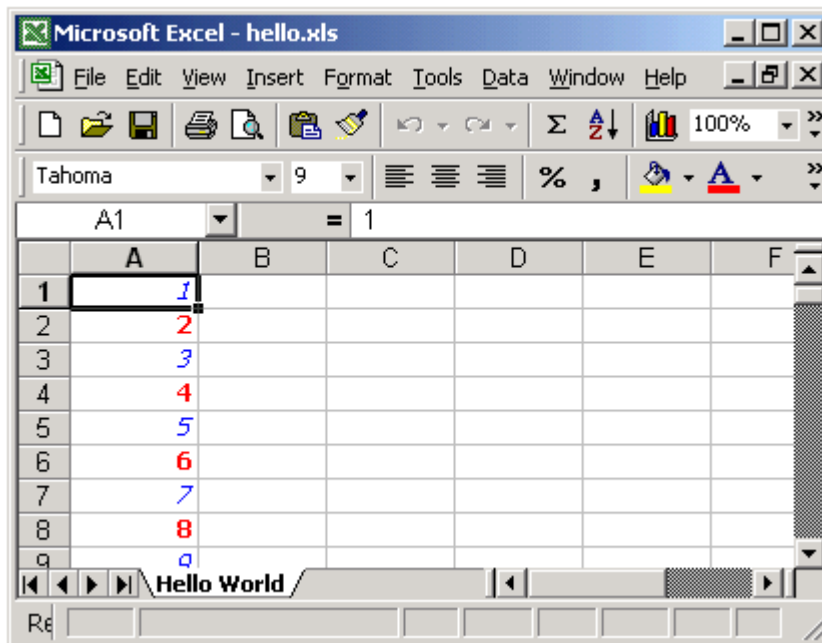
[To write code in C#](#)

C#	Copy Code
<pre> // step 1: create a new workbook C1XLBook c1XLBook1 = new C1XLBook(); // step 2: create styles for odd and even values XLStyle styleOdd = new XLStyle(c1XLBook1); styleOdd.Font = new Font("Tahoma", 9, FontStyle.Italic); styleOdd.ForeColor = Color.Blue; XLStyle styleEven = new XLStyle(c1XLBook1); styleEven.Font = new Font("Tahoma", 9, FontStyle.Bold); styleEven.ForeColor = Color.Red; // step 3: write content and styles into some cells XLSheet sheet = c1XLBook1.Sheets[0]; for (int i = 0; i < 100; i++) { XLCell cell = sheet[i, 0]; cell.Value = i + 1; cell.Style = ((i+1) % 2 == 0)? styleEven: styleOdd; } </pre>	


```
// step 4: save the file
c1XLBook1.Save(@"c:\temp\hello.xls");
```

The code is similar. The main difference is the new step 2, which creates styles for odd and even cells. The new styles are assigned to cells in step 3, along with the cell values.

This is what the file created by the code above looks like when opened in Microsoft Excel:



Worksheets

Worksheets are the individual grids contained in an Excel file. They are represented by **XLSheet** objects accessible through the **Sheets** property in the **C1XLBook** class. Each sheet has a name and contains a collection of rows and columns. Individual cells can be accessed using the **XLSheet** indexer, which takes row and column indices.

The **Rows** and **Columns** collections in the **XLSheet** object extend automatically when you use their indexers. For example, if you write the following code and the sheet has fewer than 1001 rows, new rows will be automatically added, and a valid row will be returned. The same applies to **XLColumn** and **XLCell** indexers. This is different from the behavior of most collection indexers in .NET, but it makes it very easy to create and populate **XLSheet** objects.

To write code in Visual Basic

Visual Basic	Copy Code
Dim sheet As XLSheet = C1XLBook1.Sheets(0)	

```
Dim row As XRow = sheet.Rows(1000)
```

To write code in C#

C#	Copy Code
<pre>XLSheet sheet = c1XLBook1.Sheets[0]; XRow row = sheet.Rows[1000];</pre>	

Rows and Columns

The [XLSheet](#) object contains collections of rows and columns that expose each individual row and column on the sheet. The exposed [XRow](#) and [XLColumn](#) objects allow you to assign the size (column width, row height), visibility, and style for each row and column on the sheet. If you don't assign any of these values, the sheet's defaults will be used (see the [DefaultRowHeight](#) and [DefaultColumnWidth](#) properties).

The default dimensions for **XRow** and **XLColumn** objects are 1, which means use the sheet's default values.

Cells

The [XLSheet](#) object also contains cells that can be accessed using an indexer that takes row and column indices. The cells are represented by [XLCell](#) objects that contain the cell value and style.

As with rows and columns, the cell indexer also extends the sheet automatically. For example, write:

To write code in Visual Basic

Visual Basic	Copy Code
<pre>Dim cell As XLCell = sheet(10, 10)</pre>	

To write code in C#

C#	Copy Code
<pre>XLCell cell = sheet[10,10];</pre>	

If the sheet has fewer than 11 rows and 11 columns, rows and columns will be added and a valid **XLCell** object will be returned.

Because the sheet expands automatically, this indexer will never return a **null** reference. If you want to check whether a particular cell exists on the sheet and you don't want to create the cell inadvertently, use the sheet's [GetCell](#) method instead of the indexer.

XLCell objects have a [Value](#) property that contains the cell contents. This property is of type **object** and it may contain strings, numeric, Boolean, DateTime, or null objects. Other types of objects cannot be saved into Excel files.

XLCell objects also have a [Style](#) property that defines the appearance of the cell. If the **Style** property is set to **null**, the cell is displayed using the default style. Otherwise, it should be set to an **XLStyle** object that defines the appearance of the cell (font, alignment, colors, format, and so on).

Styles

The [XLStyle](#) class defines the appearance of a cell, row, or column on a sheet. **XLStyle** includes properties that specify style elements such as the font, alignment, colors, and format used to display cell values. Not all style elements need to be defined in every **XLStyle** object. For example, if an **XLStyle** specifies only a format, then the cell is displayed using the specified format and default settings for the other style elements (font, alignment, and so on).

Excel for .NET Frequently Asked Questions

The following questions were compiled from frequently asked user questions about **Excel for .NET**.

Does Excel for .NET support creating or adding formulas or named ranges?

Yes, **Excel for .NET** supports formulas and named ranges.

Full support for binary parsing and creating formulas was added in the 2009 v2 release of the ComponentOne Studios. For example, suppose you have the following C# code:

[To write code in C#](#)

C#	Copy Code
<pre>private void Form1_Load(object sender, EventArgs e) { C1.C1Excel.XLSheet sheet = c1XLBook1.Sheets[0]; sheet[4, 1].Value = "apples and oranges"; sheet[5, 1].Value = "apples an"; // <-- result of the formula sheet[5, 1].Formula = "LEFT(B5,9)";</pre>	

```

        c1XLBook1.Save(@"c:\mybook.xls");
        System.Diagnostics.Process.Start(@"c:\mybook.xls");
    }

```

When the program runs, the resulting spreadsheet looks like the following:

	A1		fx
	A	B	C
1			
2			
3			
4			
5		apples and oranges	
6		apples an	
7			

The formula gets the information stored in column B5 and nine characters starting from the left, or "apples an".

Can Excel for .NET be used on an ASP.NET 2.0 page?

Yes it can, but there is an extra step to perform in order to be able to use **Excel for .NET** and other non GUI .NET controls in an ASP.NET 2.0 page. You must go to **View|Component Designer** from the main menu, or select "View Component Designer" from the context menu of the design view of an .aspx page. This will display an area where you can drag and drop the C1XLBook component to add it to your Web form.

Excel for .NET Task-Based Help

The task-based help assumes that you are familiar with programming in Visual Studio .NET. By following the steps outlined in the Help, you will be able to create projects demonstrating a variety of **Excel for .NET** features and get a good sense of what **Excel for .NET** can do.

Each task-based help topic also assumes that you have created a new .NET project and added the appropriate directives (**using C1.C1Excel;** for **C#**; **Imports C1.C1Excel** for **Visual Basic**) to the code.

See Also

[Adding Content to a Workbook](#)

[Merging Cells](#)

[Formatting Cells](#)

[Copying Rows from One Book to Another](#)

[Adding an Image to a Cell](#)

[Adding a Comment to a Cell](#)

[Adding a Page Break to a Worksheet](#)

[Setting the Calculation Mode for a Workbook](#)

[Importing and Exporting OpenXml Files](#)

[Creating Subtotals](#)

[Saving and Loading CSV Files](#)

Adding Content to a Workbook

To create a new workbook and add values to the first ten cells, complete the following steps:

1. Double-click the [C1XLBook](#) component in the Toolbox to add it to your form.
2. Add values to the first ten cells:

[To write code in Visual Basic](#)

Visual Basic	Copy Code
<pre>Dim sheet As XLSheet = C1XLBook1.Sheets(0) Dim i As Integer For i = 0 To 9 sheet(i,0).Value = i + 1 Next i</pre>	

[To write code in C#](#)

C#	Copy Code
<pre>XLSheet sheet = c1XLBook1.Sheets[0]; for (int i = 0; i <= 9; i++) { sheet[i,0].Value = i + 1; }</pre>	

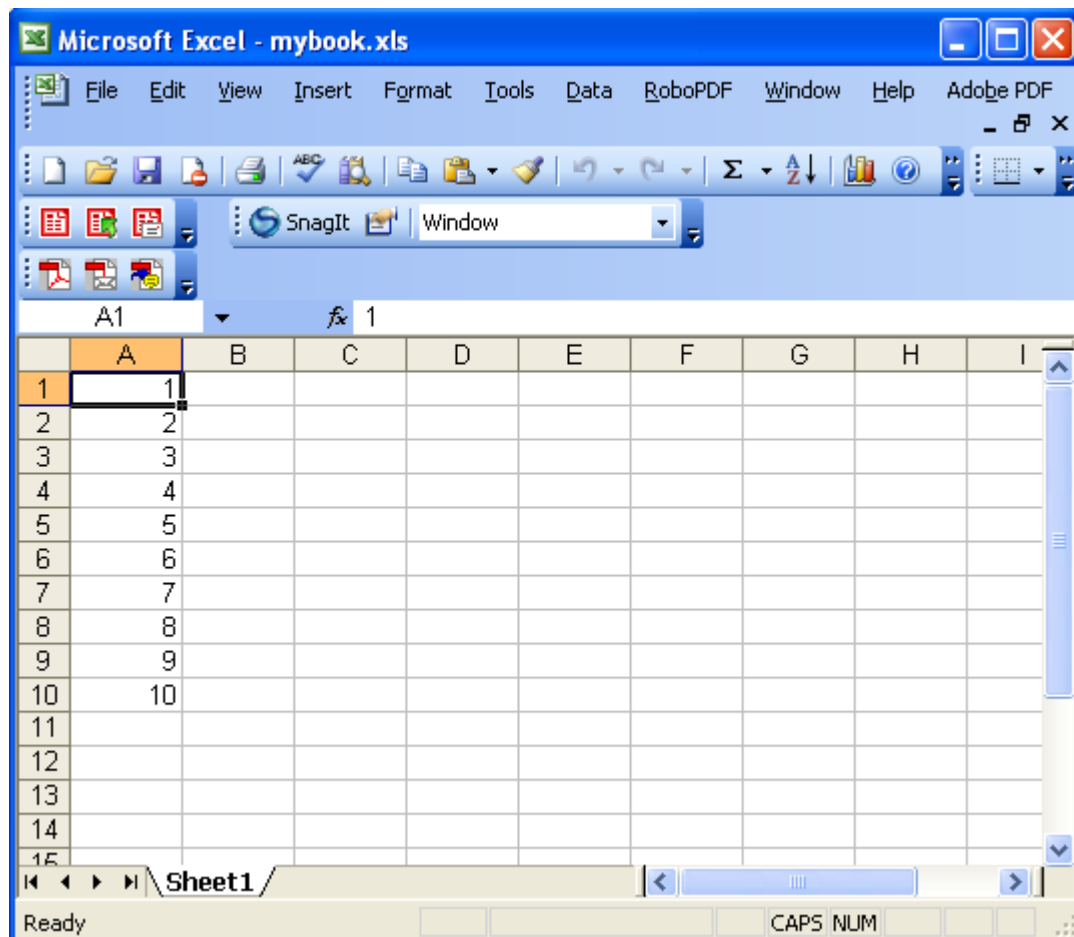
3. Save and open the book:

To write code in Visual Basic

Visual Basic	Copy Code
<pre>C1XLBook1.Save("c:\mybook.xls") System.Diagnostics.Process.Start("C:\mybook.xls")</pre>	

To write code in C#

C#	Copy Code
<pre>c1XLBook1.Save(@"c:\mybook.xls"); System.Diagnostics.Process.Start(@"C:\mybook.xls");</pre>	



Merging Cells

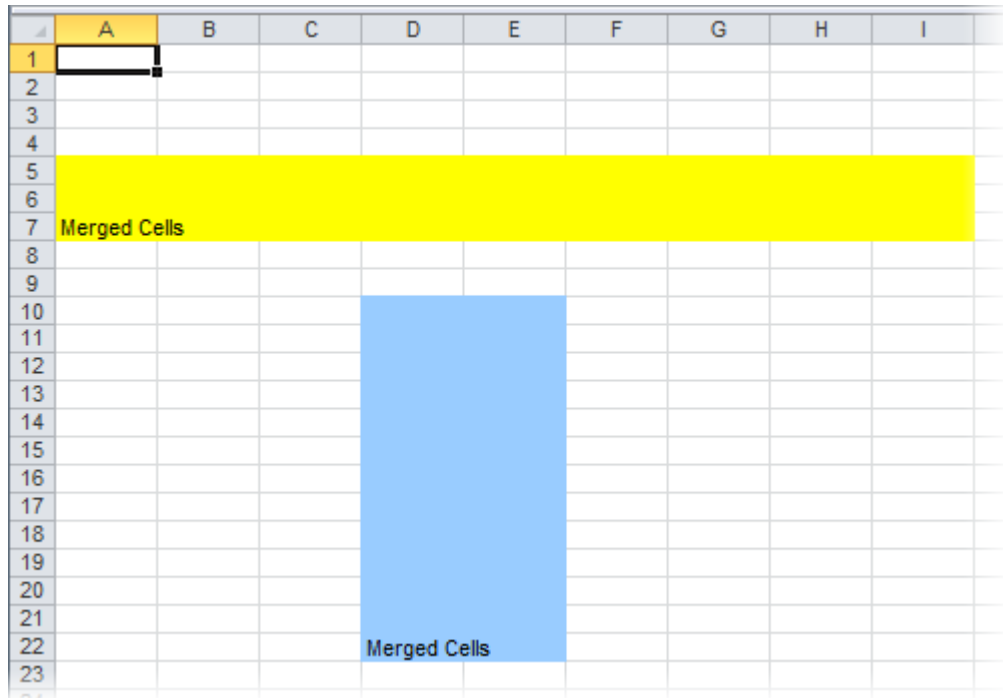
Cells can be merged using the [C1Excel.MergedCells](#) property and providing the range of cells to be merged together. Merged cells can be used for inspecting, adding, or clearing merged ranges in a sheet. Each merged range is represented by the [C1.C1Excel.XLCellRange](#) object. Cell merges are preserved when adding and removing rows or columns.

To merge cells, complete the following steps:

1. Double-click the [C1XLBook](#) component in the Toolbox to add it to your form.
2. Use the following code to create two **XLCellRanges** (**_ColRange** and **_RowRange**) and add them to the MergedCells collection. The code will also apply styles to the merged cells. You can add this code to the **Form1_Load** event, for example:

To write code in C#

C#	Copy Code
<pre>private void Form1_Load(object sender, EventArgs e) { //Select range to Merge XLCellRange _ColRange = new C1.C1Excel.XLCellRange(4, 6, 0, 8); XLCellRange _RowRange = new C1.C1Excel.XLCellRange(9, 21, 3, 4); //Set Text for Merged Cells c1XLBook1.Sheets[0][4, 0].Value = "Merged Cells"; c1XLBook1.Sheets[0][9, 3].Value = "Merged Cells"; //Merge Cells c1XLBook1.Sheets[0].MergedCells.Add(_ColRange); c1XLBook1.Sheets[0].MergedCells.Add(_RowRange); //Define and Apply Styles for Merged Cells XLStyle _Colstyle = new C1.C1Excel.XLStyle(c1XLBook1); _Colstyle.BackColor = Color.Yellow; XLStyle _Rowstyle = new XLStyle(c1XLBook1); _Rowstyle.BackColor = Color.LightBlue; c1XLBook1.Sheets[0][4, 0].Style = _Colstyle; c1XLBook1.Sheets[0][9, 3].Style = _Rowstyle; //Save and open the book c1XLBook1.Save(@"c:\mybook.xls"); System.Diagnostics.Process.Start(@"C:\mybook.xls"); }</pre>	



Formatting Cells

1. Double-click the [C1XLBook](#) component in the Toolbox to add it to your form.
2. Create a new style and assign the style some properties:

To write code in Visual Basic

Visual Basic

Copy Code

```
Dim style1 As New XLStyle(C1XLBook1)
style1.ForeColor = Color.Gold
style1.BackColor = Color.Blue
style1.Format = "$ .00"
```

To write code in C#

C#

Copy Code

```
XLStyle style1 = new XLStyle(c1XLBook1);
style1.ForeColor = Color.Gold;
style1.BackColor = Color.Blue;
style1.Format = "$ .00";
```

3. Add some content and apply the style to the cells in the first column of the sheet:

[To write code in Visual Basic](#)

Visual Basic

Copy Code

```
Dim sheet As XLSheet = C1XLBook1.Sheets(0)
Dim i As Integer
For i = 0 To 9
    sheet(i,0).Value = i + 1
    sheet(i, 0).Style = style1
Next i
```

[To write code in C#](#)

C#

Copy Code

```
C1.C1Excel.XLSheet sheet = c1XLBook1.Sheets[0];
int i;
for (i = 0; i <= 9; i++)
{
    sheet[i,0].Value = i + 1;
    sheet[i, 0].Style = style1;
}
```

4. Save and open the book:

[To write code in Visual Basic](#)

Visual Basic

Copy Code

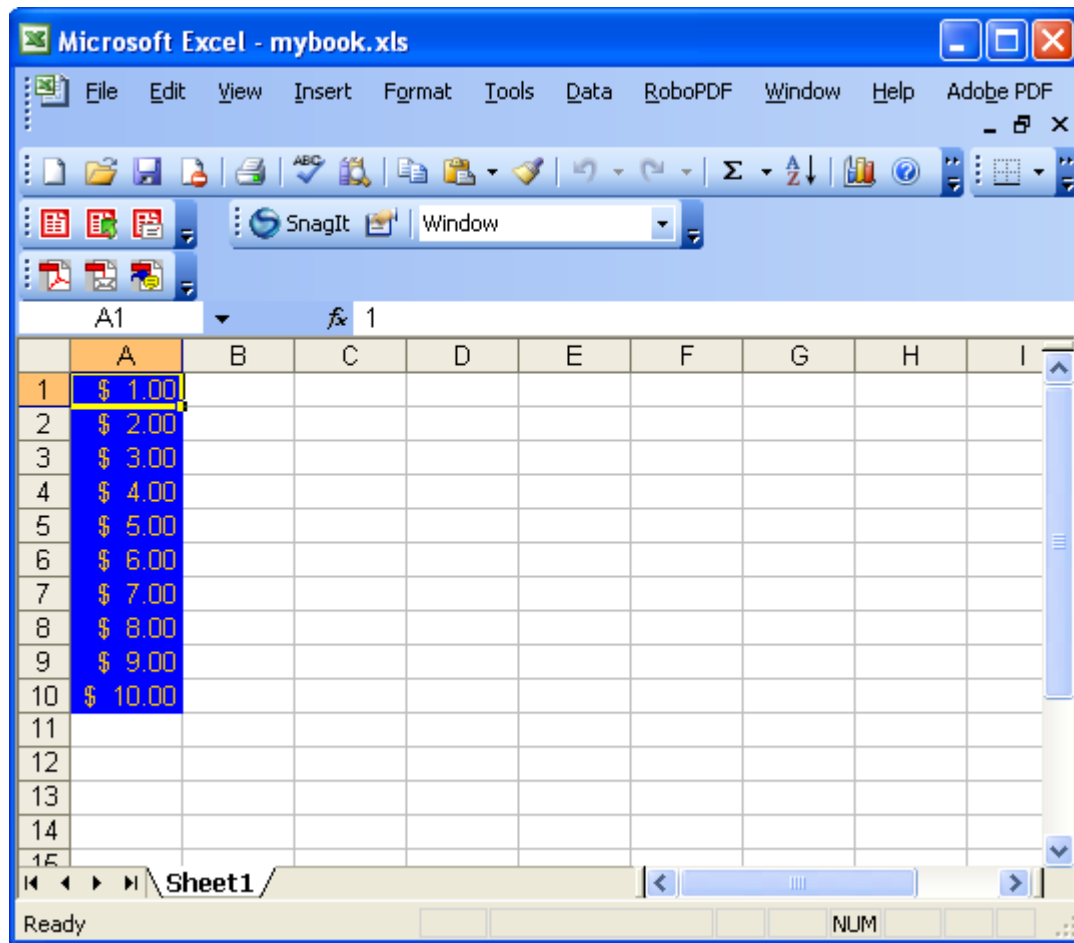
```
C1XLBook1.Save("c:\mybook.xls")
System.Diagnostics.Process.Start("C:\mybook.xls")
```

[To write code in C#](#)

C#

Copy Code

```
c1XLBook1.Save(@"c:\mybook.xls");
System.Diagnostics.Process.Start(@"C:\mybook.xls");
```



Copying Rows from One Book to Another

To copy the rows of a sheet from one book to a second book, complete the following steps:

1. Load an existing book:

To write code in Visual Basic

Visual Basic

Copy Code

```
Dim wb As New C1XLBook()
wb.Load("C:\test.xls")
```

To write code in C#

C#

Copy Code

```
C1XLBook wb = new C1XLBook();
wb.Load(@"C:\test.xls");
```

2. Create a new XLSheet:

[To write code in Visual Basic](#)

Visual Basic	Copy Code
<pre>Dim xb As New C1XLBook() xb.Sheets.Add("Test")</pre>	

[To write code in C#](#)

C#	Copy Code
<pre>C1XLBook xb = new C1XLBook(); xb.Sheets.Add("Test");</pre>	

3. Copy each row from the sheet of the existing book to new XLSheet:

[To write code in Visual Basic](#)

Visual Basic	Copy Code
<pre>Dim source As XLSheet = wb.Sheets(0) Dim dest As XLSheet = xb.Sheets("Test") Dim row As Integer, col As Integer For row = 0 To source.Rows.Count - 1 For col = 0 To source.Columns.Count - 1 dest(row, col).Value = source(row, col).Value Next col Next row</pre>	

[To write code in C#](#)

C#	Copy Code
<pre> XLSheet source = wb.Sheets[0]; XLSheet dest = xb.Sheets("Test"); for (int row = 0; row <= source.Rows.Count - 1; row++) { for (int col = 0; col <= source.Columns.Count - 1; col++) { dest[row, col].Value = source[row, col].Value; } } </pre>	

4. Save and open the new book:

[To write code in Visual Basic](#)

Visual Basic	Copy Code
<pre> xb.Save("C:\test2.xls") System.Diagnostics.Process.Start("C:\test2.xls") </pre>	

[To write code in C#](#)

C#	Copy Code
<pre> xb.Save(@"c:\test2.xls"); System.Diagnostics.Process.Start(@"C:\test2.xls"); </pre>	

Adding an Image to a Cell

Images can be added to a sheet or cell using one of the following methods. Click a link below for detailed steps on performing this task.

[Method 1: Assign an image directly to a cell's XLCell.Value property.](#)

Using this method, the image is added to the sheet and kept at its original size. The upper left corner of the image coincides with the upper left corner of the specified cell.

1. Load an existing workbook or add some content to a new workbook.

[To write code in Visual Basic](#)

Visual Basic	Copy Code
<pre>Dim wb As New C1XLBook wb.Load("C:\Project\WorkBook1.xls")</pre>	

[To write code in C#](#)

C#	Copy Code
<pre>C1XLBook wb = new C1XLBook(); wb.Load(@"C:\Project\WorkBook1.xls");</pre>	

2. Specify the image and assign it to the cell's [Value](#) property.

[To write code in Visual Basic](#)

Visual Basic	Copy Code
<pre>Dim img As Image = Image.FromFile("C:\Project\MyImage.bmp") Dim sheet As XLSheet = wb.Sheets("Forecasting Report") sheet(0, 0).Value = img</pre>	

[To write code in C#](#)

C#	Copy Code
<pre>Image img = Image.FromFile(@"C:\Project\MyImage.bmp"); XLSheet sheet = wb.Sheets("Forecasting Report"); sheet[0,0].Value = img;</pre>	

3. Save and open the new book:


To write code in Visual Basic

Visual Basic	Copy Code
<pre>wb.Save("C:\Project\WorkBook1.xls ") System.Diagnostics.Process.Start("C:\Project\WorkBook1.xls")</pre>	

To write code in C#

C#	Copy Code
<pre>wb.Save(@"C:\Project\WorkBook1.xls"); System.Diagnostics.Process.Start(@"C:\Project\WorkBook1.xls");</pre>	

In this example, the image replaces the value in the first cell, and it appears at its original size in the first cell.

	A	B	C	D
1			Project Manager:	John Smit
2			Project Completion Date:	January 11
3				
4	Schedule and Cost			
5	Budget at Completion (BAC)	\$2,000		Estimate to
6	Earned Value (EV)	\$1,050		Estimate at
7	Actual Cost to Date (AC)	\$950		Cost Variat
8	Planned Value (PV)	\$10,000		Schedule \
9				Cost Perfor

Method 2: Create an `XLPictureShape` object, set its properties, and assign it to a cell's `XLCell.Value` property.

This second method allows you to customize the image by specifying its size, rotation angle, brightness, contrast, border, and more.

1. Load an existing workbook or add some content to a new workbook.

To write code in Visual Basic

Visual Basic	Copy Code
<pre>Dim wb As New C1XLBook wb.Load("C:\Project\WorkBook1.xls")</pre>	

To write code in C#

C#	Copy Code
<pre>C1XLBook wb = new C1XLBook(); wb.Load(@"C:\Project\WorkBook1.xls");</pre>	

2. Create an [XLPictureShape](#) object, set some of its properties and assign it to a cell's Value property.

To write code in Visual Basic

Visual Basic	Copy Code
<pre>Dim img As Image = Image.FromFile("C:\Project\MyImage.bmp") Dim pic As New XLPictureShape(img, 1500, 1500) pic.Rotation = 30.0F pic.LineColor = Color.DarkRed pic.LineWidth = 100 ' assign the pic to the first cell of the specified sheet Dim sheet As XLSheet = wb.Sheets("Forecasting Report") sheet(0, 0).Value = pic</pre>	

To write code in C#

C#	Copy Code
<pre>Image img = Image.FromFile(imageFileName); XLPictureShape pic = new XLPictureShape(img, 1500, 1500); pic.Rotation = 30.0f;</pre>	

```

pic.LineColor = Color.DarkRed;
pic.LineWidth = 100;
// assign the pic to the first cell of the specified sheet
XLSheet sheet = wb.Sheets("Forecasting Report");
sheet[0,0].Value = pic;

```

3. Save and open the book.

To write code in Visual Basic

Visual Basic	Copy Code
<pre> wb.Save("C:\Project\WorkBook1.xls ") System.Diagnostics.Process.Start("C:\Project\WorkBook1.xls") </pre>	

To write code in C#

C#	Copy Code
<pre> wb.Save(@"C:\Project\WorkBook1.xls"); System.Diagnostics.Process.Start(@"C:\Project\WorkBook1.xls"); </pre>	

In this example, the image replaces the value in the first cell, is rotated 30°, and has a dark red border. Since we have specified the horizontal and vertical position of the image, it does not appear in the first cell.

	A	B	C
1			Project Manager:
2	Department: Development		Project Completio
3			
4	Schedule and Cost		
5	Budget at Completion (BAC)	0.000	
6	Earned Value (EV)		
7	Actual Cost to Date (AC)		
8	Planned Value (PV)		
9			
10			
11			
12	Milestones		



Method 3: Create an [XLPictureShape](#) object, set its properties, and add it to a sheet's [ShapeCollection](#).

This method uses the [XLPictureShape](#) constructor to specify the image boundaries in sheet coordinates. The shape is added directly to the sheet's [ShapeCollection](#), rather than to a specific cell.

1. Load an existing workbook or add some content to a new workbook.

To write code in Visual Basic

Visual Basic	Copy Code
<pre>Dim wb As New C1XLBook wb.Load("C:\Project\WorkBook1.xls")</pre>	

To write code in C#

C#	Copy Code
<pre>C1XLBook wb = new C1XLBook(); wb.Load(@"C:\Project\WorkBook1.xls");</pre>	

2. Create an [XLPictureShape](#) object, set some of its properties and assign it to a sheet's [ShapeCollection](#).

To write code in Visual Basic

Visual Basic	Copy Code
<pre>Dim img As Image = Image.FromFile("C:\Project\MyImage.bmp") Dim pic As New XLPictureShape(img, 3000, 3500, 2500, 900) pic.Rotation = 30.0F pic.LineColor = Color.DarkRed pic.LineWidth = 100 ' add the pic to specified sheet's ShapeCollection Dim sheet As XLSheet = wb.Sheets("Forecasting Report") sheet.Shapes.Add(pic)</pre>	

To write code in C#

C#	Copy Code
<pre>Image img = Image.FromFile(@"C:\Project\MyImage.bmp"); XLPictureShape pic = new XLPictureShape(img, 3000, 3500, 2500, 900); pic.Rotation = 30.0f; pic.LineColor = Color.DarkRed; pic.LineWidth = 100; // add the pic to specified sheet's ShapeCollection XLSheet sheet = wb.Sheets("Forecasting Report"); sheet.Shapes.Add(pic)</pre>	

3. Save and open the book.

To write code in Visual Basic

Visual Basic	Copy Code
<pre>wb.Save("C:\Project\WorkBook1.xls ") System.Diagnostics.Process.Start("C:\Project\WorkBook1.xls")</pre>	

To write code in C#

C#	Copy Code
<pre>wb.Save(@"C:\Project\WorkBook1.xls"); System.Diagnostics.Process.Start(@"C:\Project\WorkBook1.xls");</pre>	

In this example, the shape was added to the sheet's [ShapeCollection](#); therefore, the image does not replace the value in the first cell. Here we specified the height and width of the image, as well as the horizontal and vertical positioning.

	A	B	C
1	Project Name: Project 3X		Project Mana
2	Department: Development		Project Comp
3			
4	Schedule and Cost		
5	Budget at Completion (BAC)	\$2,000	
6	Earned Value (EV)	\$1,050	
7	Actual Cost to Date (AC)	\$950	
8	Planned Value (PV)	\$10,000	
9			
10			
11			
12	Milestones		
13	Milestones		
14	Milestone A: Charter sign-off		
15	Milestone B: Business requirement sign-off		
16	Milestone C: Functional spec. sign-off		
17	Milestone D: Phase gate review		
18	Scope		



Adding a Comment to a Cell

1. Double-click the [C1XLBook](#) component in the Toolbox to add it to your form.
2. Add text to a cell in the worksheet using the following code:

To write code in Visual Basic

Visual Basic

Copy Code

```
C1XLBook1.Sheets(0)(2, 3).Value = "test"
```

To write code in C#

C#

Copy Code

```
c1XLBook1.Sheets[0][2, 3].Value = "test";
```

3. Add a comment to the [XLCommentCollection](#) and create a box to show it in using the following code:

To write code in Visual Basic

Visual Basic**Copy Code**

```
C1XLBook1.Sheets(0).Comments.Add(2, 3, "John", "Test comment")  
C1XLBook1.Sheets(0).Comments(0).TextBox.Rectangle = New Rectangle(220, 210, 1900,  
1200)
```

[To write code in C#](#)

C#**Copy Code**

```
c1XLBook1.Sheets[0].Comments.Add(2, 3, "John", "Test comment");  
c1XLBook1.Sheets[0].Comments[0].TextBox.Rectangle = new Rectangle(220, 210, 1900,  
1200);
```

4. Save and open the book:

[To write code in Visual Basic](#)

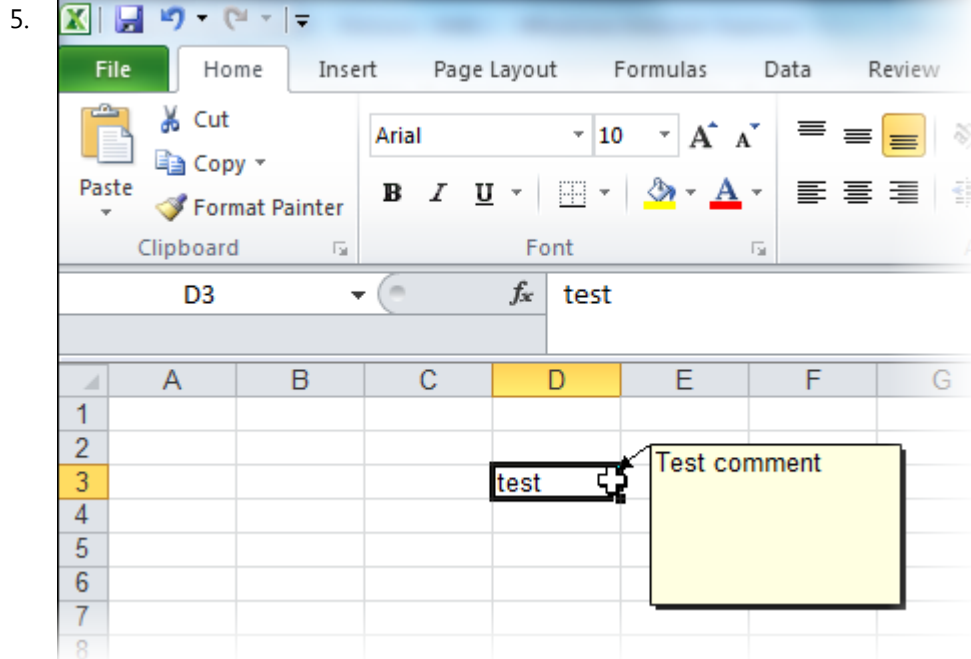
Visual Basic**Copy Code**

```
C1XLBook1.Save("c:\mybook.xls")  
System.Diagnostics.Process.Start("C:\mybook.xls")
```

[To write code in C#](#)

C#**Copy Code**

```
c1XLBook1.Save(@"c:\mybook.xls");  
System.Diagnostics.Process.Start(@"C:\mybook.xls");
```



Adding a Page Break to a Worksheet

You can easily add page breaks in rows and columns for files in OpenXML (.xlsx) format using the [PageBreak](#) and [PageBreak](#) properties.

1. Double-click the [C1XLBook](#) component in the Toolbox to add it to your form.
2. Add some text values and page breaks using the following code:

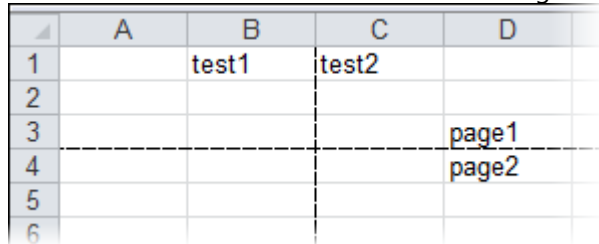
To write code in Visual Basic

Visual Basic	Copy Code
<pre> C1XLBook1.Sheets(0)(2, 3).Value = "page1" C1XLBook1.Sheets(0).Rows(2).PageBreak = True C1XLBook1.Sheets(0)(0, 1).Value = "test1" C1XLBook1.Sheets(0)(0, 2).Value = "test2" C1XLBook1.Sheets(0).Columns(1).PageBreak = True C1XLBook1.Sheets(0)(3, 3).Value = "page2" ' Save and open the .xlsx file C1XLBook1.Save("c:\Save.xlsx") System.Diagnostics.Process.Start("c:\Save.xlsx") </pre>	

To write code in C#

C#	Copy Code
<pre>c1XLBook1.Sheets[0][2, 3].Value = "page1"; c1XLBook1.Sheets[0].Rows[2].PageBreak = true; c1XLBook1.Sheets[0][0, 1].Value = "test1"; c1XLBook1.Sheets[0][0, 2].Value = "test2"; c1XLBook1.Sheets[0].Columns[1].PageBreak = true; c1XLBook1.Sheets[0][3, 3].Value = "page2"; // Save and open the .xlsx file c1XLBook1.Save(@"c:\Save.xlsx"); System.Diagnostics.Process.Start(@"c:\Save.xlsx");</pre>	

3. Run the project to open the .xlsx file.
4. In Excel, select the **Page Layout** tab, and select the **Print** checkbox under **Gridlines**. The worksheet should look similar to the following:



The screenshot shows an Excel worksheet with columns A, B, C, and D, and rows 1 through 6. A vertical dashed line is between columns B and C, and a horizontal dashed line is between rows 3 and 4. The cell B1 contains 'test1', C1 contains 'test2', D3 contains 'page1', and D4 contains 'page2'.

	A	B	C	D
1		test1	test2	
2				
3				page1
4				page2
5				
6				

Setting the Calculation Mode for a Workbook

The [CalculationMode](#) property specifies the calculation mode for all formulas in the workbook. The [CalculationMode](#) enumeration provides three options: **Manual** (you manually perform the calculation), **Auto** (the calculation is automatically performed), or **AutoNoTable** (the calculation is performed except on tables).

To set the calculation mode, follow these steps:

1. Double-click the [C1XLBook](#) component in the Toolbox to add it to your form.
2. Add a simple formula using the following code:

To write code in Visual Basic

Visual Basic	Copy Code
<pre>Dim sheet As XLSheet = c1XLBook1.Sheets(0)</pre>	

```
' simple formula
sheet(7, 0).Value = "Formula: 5!"
sheet(7, 1).Value = 122
sheet(7, 1).Formula = "1*2*3*4*5"
c1XLBook1.CalculationMode = CalculationMode.Auto
c1XLBook1.Save("c:\Save.xlsx")
System.Diagnostics.Process.Start("c:\Save.xlsx")
```

To write code in C#

C#	Copy Code
<pre>XLSheet sheet = c1XLBook1.Sheets[0]; // simple formula sheet[7, 0].Value = "Formula: 5!"; sheet[7, 1].Value = 122; sheet[7, 1].Formula = "1*2*3*4*5"; c1XLBook1.CalculationMode = CalculationMode.Auto; c1XLBook1.Save(@"c:\Save.xlsx"); System.Diagnostics.Process.Start(@"c:\Save.xlsx");</pre>	

- Run the project to open the Excel file. Notice that the value for the cell in (7,1) is **120**, or the total of **1*2*3*4*5**, not **122**, since we set the CalculationMode to **Auto**.

	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					
8	Formula: 5!	120			
9					

Importing and Exporting OpenXml Files

ComponentOne Excel for .NET can now read and write Microsoft Excel 2007 OpenXml files. OpenXml is an open, standards-based format introduced by Microsoft in Office 2007. OpenXml files are easier to manipulate by applications because OpenXml is based on XML and is publicly documented, as opposed to proprietary binary formats, such as BIFF8. OpenXml files contain a number of XML files compressed using Zip compression. Because they are compressed, OpenXml files are usually much smaller than traditional document files, such as .doc and .xls files.

Excel for .NET can load and save data and formatting information in OpenXml files; however, formulas are not loaded or saved. They are copied in BIFF format as opaque, which is not supported by the OpenXML format at this time. If you load files containing formulas and then save them, the formulas will be removed. This is in contrast to the traditional .xls, or BIFF8, format, which preserves the formulas.

To support the OpenXml format, the [C1XLBook Load](#) and [Save](#) methods received overloads that take a **FileFormat** parameter that is used to specify the file format to use when loading or saving files.

If the file name is not specified, then **Excel for .NET** infers the file format from the file name extension: files with an "XLSX" and "ZIP" extension are loaded and saved as OpenXml files, by default. Other files are loaded and saved as BIFF8, or .xls, format.

For example:

[To write code in C#](#)

C#	Copy Code
<pre>//load and save relying on file extension c1Excel1.Load("somefile.xls"); // load biff 8 file c1Excel1.Save("somefile.xlsx"); // save file as OpenXml c1Excel1.Save("somefile.zip"); // save file as OpenXml // load and save specifying the FileFormat c1Excel1.Load("somefile.xls", FileFormat.Biff8); c1Excel1.Save("somefile.xlsx", FileFormat.OpenXml);</pre>	

You can also specify the format when loading or saving files to and from streams. If the **FileFormat** is not specified, then **Excel for .NET** uses the BIFF8 format as a default.

Note that there is a small behavior change implied here. Consider the statement below:

```
C1Excel1.Save("somefile.xlsx");
```


In previous versions of **Excel for .NET**, this would save a BIFF8 file (with the wrong extension). Now, this will save an OpenXml file (with the correct extension). If you have code like this in your applications, you should change it to the following when upgrading:

To write code in C#

C#	Copy Code
<pre>// deliberately save file with wrong extension C1Excel1.Save("somefile.xlsx", FileFormat.Biff8);</pre>	

To export a book to an OpenXml file, complete the following steps:

1. Load an existing book:

To write code in Visual Basic

Visual Basic	Copy Code
<pre>Dim wb As New C1XLBook() wb.Load("C:\test.xlsx") ' or Dim wb As New C1XLBook() wb.Load("C:\test.xlsx", C1.C1Excel.FileFormat.OpenXml)</pre>	

To write code in C#

C#	Copy Code
<pre>C1XLBook wb = new C1XLBook(); wb.Load(@"C:\test.xlsx"); // or C1XLBook wb = new C1XLBook(); wb.Load(@"C:\test.xlsx", C1.C1Excel.FileFormat.OpenXml);</pre>	

2. Export the book to an OpenXml Format file:

To write code in Visual Basic

Visual Basic	Copy Code
<pre>Dim wb As New C1XLBook() ' Add some content Dim sheet As XLSheet = wb.Sheets(0) Dim i As Integer For i = 0 To 9 sheet(i,0).Value = i + 1 Next i ' Export to OpenXml Format file wb.Save("C:\test.xlsx") ' or ' Export to OpenXml Format file wb.Save("C:\test.xlsx", C1.C1Excel.FileFormat.OpenXml)</pre>	

To write code in C#

C#	Copy Code
<pre>C1XLBook wb = new C1XLBook(); // Add some content XLSheet sheet = wb.Sheets[0]; for (int i = 0; i <= 9; i++) { sheet[i,0].Value = i + 1; } // Export to OpenXml Format file wb.Save(@"C:\test.xlsx"); // or // Export to OpenXml Format file wb.Save(@"C:\test.xlsx", C1.C1Excel.FileFormat.OpenXml);</pre>	

Creating Subtotals

The following code provides an example of how to format the cells of a book.

1. Double-click the [C1XLBook](#) component in the Toolbox to add it to your form.
2. Select **View | Code** and add one of the following statements at the top of the form:
 - Import C1.C1Excel (Visual Basic)
 - using C1.C1Excel; (C#)
3. Add the following code to the **Form_Load** event:

To write code in Visual Basic

Visual Basic	Copy Code
<pre> Private Sub Form1_Load(sender As Object, e As EventArgs) Dim book As New C1XLBook() Dim sheet As XLSheet = book.Sheets(0) Dim totalStyle As New XLStyle(book) totalStyle.Font = New Font(book.DefaultFont, FontStyle.Bold) sheet(2, 1).Value = "Number" sheet(2, 2).Value = "ID" sheet(3, 1).Value = 12 sheet(3, 2).Value = 17 sheet.Rows(3).OutlineLevel = 2 sheet.Rows(3).Visible = False sheet(4, 1).Value = 12 sheet(4, 2).Value = 14 sheet.Rows(4).OutlineLevel = 2 sheet.Rows(4).Visible = False sheet(5, 1).Value = "12 Total" sheet(5, 1).Style = totalStyle sheet(5, 2).Value = 31 sheet(5, 2).Formula = "SUBTOTAL(9,C4:C5)" sheet.Rows(5).OutlineLevel = 1 sheet(6, 1).Value = 34 sheet(6, 2).Value = 109 sheet.Rows(6).OutlineLevel = 2 sheet(7, 1).Value = "34 Total" sheet(7, 1).Style = totalStyle sheet(7, 2).Value = 109 sheet(7, 2).Formula = "SUBTOTAL(9,C7:C7)" sheet.Rows(7).OutlineLevel = 1 sheet(8, 1).Value = "Grand Total" </pre>	

```

sheet(8, 1).Style = totalStyle
sheet(8, 2).Value = 140
sheet(8, 2).Formula = "SUBTOTAL(9,C4:C7)"
sheet.Rows(8).OutlineLevel = 0
book.Save("c:\mybook.xls")
System.Diagnostics.Process.Start("C:\mybook.xls")
End Sub

```

To write code in C#

C#	Copy Code
<pre> private void Form1_Load(object sender, EventArgs e) { C1XLBook book = new C1XLBook(); XLSheet sheet = book.Sheets[0]; XLStyle totalStyle = new XLStyle(book); totalStyle.Font = new Font(book.DefaultFont, FontStyle.Bold); sheet[2, 1].Value = "Number"; sheet[2, 2].Value = "ID"; sheet[3, 1].Value = 12; sheet[3, 2].Value = 17; sheet.Rows[3].OutlineLevel = 2; sheet.Rows[3].Visible = false; sheet[4, 1].Value = 12; sheet[4, 2].Value = 14; sheet.Rows[4].OutlineLevel = 2; sheet.Rows[4].Visible = false; sheet[5, 1].Value = "12 Total"; sheet[5, 1].Style = totalStyle; sheet[5, 2].Value = 31; sheet[5, 2].Formula = "SUBTOTAL(9,C4:C5)"; sheet.Rows[5].OutlineLevel = 1; sheet[6, 1].Value = 34; sheet[6, 2].Value = 109; sheet.Rows[6].OutlineLevel = 2; sheet[7, 1].Value = "34 Total"; sheet[7, 1].Style = totalStyle; </pre>	

```

        sheet[7, 2].Value = 109;
        sheet[7, 2].Formula = "SUBTOTAL(9,C7:C7)";
        sheet.Rows[7].OutlineLevel = 1;
        sheet[8, 1].Value = "Grand Total";
        sheet[8, 1].Style = totalStyle;
        sheet[8, 2].Value = 140;
        sheet[8, 2].Formula = "SUBTOTAL(9,C4:C7)";
        sheet.Rows[8].OutlineLevel = 0;
        book.Save(@"c:\mybook.xls");
        System.Diagnostics.Process.Start(@"C:\mybook.xls");
    }

```

4. Run the program. The spreadsheet will open and look similar to the following:

1	2	3	A	B	C
1					
2					
3			Number	ID	
4				12	17
5				12	14
6			12 Total		31
7				34	109
8			34 Total		109
9			Grand Total		140
10					

The SUBTOTAL formulas get the sum of the specified rows.

Saving and Loading CSV Files

ComponentOne Excel for .NET supports saving and loading comma-separated values (CSV) files. CSV is a common file format that stores tabular data, including numbers and text, in plain text form for easy readability.

The following code provides an example of how to save and load a .csv file.

1. Double-click the [C1XLBook](#) component in the Toolbox to add it to your form.
2. Select **View | Code** and add one of the following statements at the top of the form:
 - Import C1.C1Excel (Visual Basic)
 - using C1.C1Excel; (C#)
3. Add the following code to the **Form_Load** event to create a sheet with 10 values and save the workbook to .csv format:

To write code in Visual Basic

Visual Basic	Copy Code
<pre> Private Sub Form1_Load(sender As Object, e As EventArgs) Dim sheet As XLSheet = c1XLBook1.Sheets(0) For i As Integer = 0 To 9 sheet(i, 0).Value = i + 1 sheet(i, 1).Value = 10 - i; Next sheet.SaveCsv("c:\test.csv") System.Diagnostics.Process.Start("C:\test.csv") End Sub </pre>	

To write code in C#

C#	Copy Code
<pre> private void Form1_Load(object sender, EventArgs e) { XLSheet sheet = c1XLBook1.Sheets[0]; for (int i = 0; i <= 9; i++) { sheet[i, 0].Value = i + 1; sheet[i, 1].Value = 10 - i; } sheet.SaveCsv(@"c:\test.csv"); System.Diagnostics.Process.Start(@"C:\test.csv"); } </pre>	

4.  ject and view the .csv file:

test.csv - Notepad
File Edit Format
1,10
2,9
3,8
4,7
5,6
6,5
7,4
8,3
9,2
10,1

5. Add some new values to the test.csv file. You'll need to save again in order to save the new values to the file. You can do this by adding code for the [LoadCsv](#) and [SaveCsv](#) methods to the **Form1_Load** event so it now looks like this:

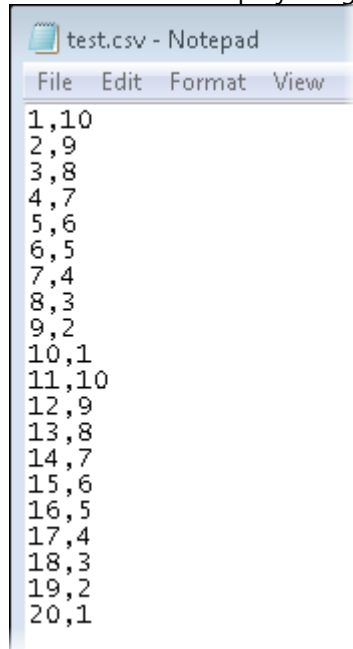
[To write code in Visual Basic](#)

Visual Basic	Copy Code
<pre>Private Sub Form1_Load(sender As Object, e As EventArgs) Dim sheet As XLSheet = c1XLBook1.Sheets(0) For i As Integer = 0 To 9 sheet(i, 0).Value = i + 1 sheet(i, 1).Value = 10 - i Next sheet.SaveCsv("c:\test.csv") sheet.LoadCsv("c:\test.csv") For i As Integer = 10 To 19 sheet(i, 0).Value = i + 1 sheet(i, 1).Value = 10 - i Next sheet.SaveCsv("c:\test.csv") System.Diagnostics.Process.Start("C:\test.csv") End Sub</pre>	

[To write code in C#](#)

C#	Copy Code
<pre> private void Form1_Load(object sender, EventArgs e) { XLSheet sheet = c1XLBook1.Sheets[0]; for (int i = 0; i <= 9; i++) { sheet[i, 0].Value = i + 1; sheet[i, 1].Value = 10 - i; } sheet.SaveCsv(@"c:\test.csv"); sheet.LoadCsv(@"c:\test.csv"); for (int i = 10; i <= 19; i++) { sheet[i, 0].Value = i + 1; sheet[i, 1].Value = 20 - i; } sheet.SaveCsv(@"c:\test.csv"); System.Diagnostics.Process.Start(@"C:\test.csv"); } </pre>	

6. Press **F5** to run the project again and view the .csv file:



Excel for .NET Samples

Please be advised that this ComponentOne software tool is accompanied by various sample projects and/or demos which may make use of other development tools included with the ComponentOne Studios.

Samples can be accessed from the **ComponentOne Sample Explorer**. To view samples, on the desktop, click the **Start** button and then click **All Programs | ComponentOne | Studio for WinForms (or Studio for ASP.NET) | Samples | Excel Samples**.

Click one of the following links to view a list of **Excel for .NET** samples:

[To write code in Visual Basic](#)

Sample	Description
AutoSizeColumns	The sample loops through all cells measuring their contents, then sets the column widths based on the widest entry. This sample takes into account the cell contents, format and font. It does not account for content wrapping or merging. This sample uses the C1XLBook component.
CellBorders	The sample allows you to create borders around arbitrary ranges. You can control border width, style, color, and so on. When you are done creating the borders, click the last button on the toolbar to export the grid to Excel, including the custom borders. This sample uses the C1FlexGrid, C1XLBook, and the Microsoft ImageList components.
CombineSheets	This sample scans all the .xls files in a folder, clones the first sheet, renames it with the file name, and adds the cloned sheets to a master workbook. It then saves the combined workbook into a file and opens that. This sample uses the C1XLBook component.
ExcelPictures	This sample creates a workbook with several sheets: "Images" shows a random collection of images. "Types" shows images of different types. "Borders" shows images with different border styles. "Alignment" shows how to align and scale images within cells. "Properties" shows the effect of additional image properties such as brightness, grayscale, and so on. This sample uses the C1XLBook component.

FlexGridExcel	Demonstrates how to load and save XLS files. This sample uses the C1FlexGrid and C1XLBook components.
HelloWorld	This sample actually does more than save a "Hello World" document. It shows how to create workbooks, assign names to the worksheets, create styles and assign them to cells, assign content to cells, and save workbooks.. This sample uses the C1XLBook component.
PrintSettings	This sample has a Load button that opens an existing XLS file and shows the print settings for the first sheet. You can modify the print settings, then click the Save button to save a new file (in the c:\temp folder). The new file is then displayed in Excel . This sample uses the C1XLBook component.

To write code in C#

Sample	Description
AutoSizeColumns	The sample loops through all cells measuring their contents, then sets the column widths based on the widest entry. This sample takes into account the cell contents, format and font. It does not account for content wrapping or merging. This sample uses the C1XLBook component.
CellBorders	The sample allows you to create borders around arbitrary ranges. You can control border width, style, color, and so on When you are done creating the borders, click the last button on the toolbar to export the grid to Excel, including the custom borders. This sample uses the C1FlexGrid, C1XLBook, and the Microsoft ImageList components.
ColorPalette	This sample shows the Excel color palette, including the colors and their RGB value. It uses both the C1XLBook and C1FlexGrid components. Note that this sample is not available at http://helpcentral.componentone.com/Samples.aspx .
CombineSheets	This sample scans all the .xls files in a folder, clones the first sheet, renames it with the file name, and adds the cloned sheets to a master workbook. It then saves the combined workbook into a file and opens that. This sample uses the

	C1XLBook component.
FlexGrid	Demonstrates how to load and save XLS files. This sample uses the C1FlexGrid and C1XLBook components.
HelloWorld	This sample actually does more than save a "Hello World" document. It shows how to create workbooks, assign names to the worksheets, create styles and assign them to cells, assign content to cells, and save workbooks.. This sample uses the C1XLBook component.
PrintSettings	This sample has a Load button that opens an existing XLS file and shows the print settings for the first sheet. You can modify the print settings, then click the Save button to save a new file (in the c:\temp folder). The new file is then displayed in Excel . This sample uses the C1XLBook component.

API Reference

The following topics contain the API reference for ComponentOne Excel for .NET.

See Also

[C1.C1Excel.4 Assembly](#)

C1.C1Excel.4 Assembly

Overview

Namespaces












Namespace	Description
C1.C1Excel	
C1.Win.Localization	













Namespaces

C1.C1Excel Namespace


Overview

















Classes



Class	Description
 C1XLBook	Represents an Excel workbook containing one or more worksheets. Contains methods for loading and saving XLS files, and exposes a collection of XLSheet objects that represent the individual worksheets.
 ShapeCollection	Represents a collection of XLShape objects on an XLSheet .
 Strings	Static class containing UI strings used by the designer.
 XLCell	Represents individual cells in an XLSheet and provides properties for getting and setting the cell Value , Style , and Hyperlink .
 XLCellRange	Represents a range of XLCell objects in an XLSheet .
 XLCellRangeCollection	Represents a collection of XLCellRange objects on an XLSheet .
 XLColumn	Represents a column in a worksheet. This class provides properties for setting the column's width, style, and visibility.
 XLColumnCollection	Represents a collection of XLColumn objects that represent the individual columns in each XLSheet .
 XLComment	Represents a comment associated with a cell.
 XLCommentCollection	Represents a collection of XLComment objects in a XLSheet .
 XLCommentShape	Represents a text shape embedded in an XLSheet .

	XLNamedRange	Represents a named range of XLCell objects on one or several worksheets.
	XLNamedRangeCollection	Represents a collection of XLNamedRange objects.
	XLOpaqueShape	Represents a shape that was inserted in the sheet using Excel and is preserved but not fully exposed by the C1XLBook component.
	XLPictureShape	Represents an Image embedded in an XLSheet .
	XLPrintSettings	Provides options and settings for printing XLSheet objects.
	XLRow	Represents a row in a worksheet, provides properties for setting the row's height, style, and visibility.
	XLRowCollection	Represents a collection of XLRow objects that represent the individual rows in each XLSheet .
	XLShape	Represents a shape embedded in an XLSheet .
	XLSheet	Represents individual worksheets in an Excel workbook (C1XLBook).
	XLSheetCollection	Collection of XLSheet objects that represent the individual worksheets in a C1XLBook .
	XLStyle	Contains style elements used to define the appearance of the cells.
	XLTextShape	Represents a common text shape embedded in an XLSheet .

Enumerations

	Enumeration	Description
	CalculationMode	Specifies calculation mode for all formulas in the workbook.

 CompatibilityMode	Specifies limits to sheet size that correspond to specific versions of Microsoft Excel.
 ConsolidationFunction	Specifies consolidation function for sheet subtotals.
 FileFormat	Specifies the file format to use when loading or saving workbooks.
 ImageScaling	Specifies how images are scaled within spreadsheet cells.
 XLAlignHorzEnum	Specifies how to align cell content horizontally within a cell.
 XLAlignVertEnum	Specifies how to align cell content vertically within a cell.
 XLDiagonalFlags	Specifies the type of diagonal line to draw across the cells.
 XLHorizTextAlign	Specifies horizontal text alignment.
 XLLineStyleEnum	Specifies the line style used for the cell borders.
 XLPatternEnum	Specifies the pattern used to fill the cell background.
 XLPictureViewType	Specifies how images should be displayed (color, grayscale, or black and white).
 XLRangeType	The range type.
 XLReferenceMode	The reference mode.
 XLReferenceType	The reference type.
 XLShapeDashedLineStyleEnum	Specifies the dash style of borders drawn around XLShape objects.
 XLShapeLineStyleEnum	Specifies the style of borders drawn around XLShape objects.

 XLTextOrientation	Specifies text orientation.
 XLVertTextAlign	Specifies vertical text alignment.

See Also

Reference

[C1.C1Excel.4 Assembly](#)

Classes

C1XLBook

Represents an Excel workbook containing one or more worksheets.

Contains methods for loading and saving XLS files, and exposes a collection of [XLSheet](#) objects that represent the individual worksheets.

Object Model

C1XLBook

Syntax

Visual Basic (Declaration)	
<pre> <System.ComponentModel.DesignerAttribute(DesignerBaseTypeName="System.ComponentModel.Design.IDesigner", DesignerTypeName="C1.C1Excel.Design.C1XLBookDesigner, C1.C1Excel.4.Design, Version=4.1.20133.280")> <System.ComponentModel.LicenseProviderAttribute(System.ComponentModel.LicenseProvider)> <System.Drawing.ToolboxBitmapAttribute()> Public Class C1XLBook Inherits System.ComponentModel.Component </pre>	
C#	
<pre> [System.ComponentModel.DesignerAttribute(DesignerBaseTypeName="System.ComponentModel.Design.IDesigner", DesignerTypeName="C1.C1Excel.Design.C1XLBookDesigner, C1.C1Excel.4.Design, Version=4.1.20133.280")] [System.ComponentModel.LicenseProviderAttribute(System.ComponentModel.LicensePro </pre>	

```
vider)]  
[System.Drawing.ToolboxBitmapAttribute()]  
public class C1XLBook : System.ComponentModel.Component
```

Inheritance Hierarchy

```
System.Object  
    System.MarshalByRefObject  
        System.ComponentModel.Component  
            C1.C1Excel.C1XLBook
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents an Excel workbook containing one or more worksheets.

Contains methods for loading and saving XLS files, and exposes a collection of [XLSheet](#) objects that represent the individual worksheets.

Object Model

C1XLBook

Syntax

Visual Basic (Declaration)

```
<System.ComponentModel.DesignerAttribute(DesignerBaseTypeName="System.ComponentModel.Design.IDesigner", DesignerTypeName="C1.C1Excel.Design.C1XLBookDesigner, C1.C1Excel.4.Design, Version=4.1.20133.280")>  
<System.ComponentModel.LicenseProviderAttribute(System.ComponentModel.LicensePro
```



```

vider)>
<System.Drawing.ToolboxBitmapAttribute()>
Public Class C1XLBook
    Inherits System.ComponentModel.Component

```

C#

```

[System.ComponentModel.DesignerAttribute(DesignerBaseTypeName="System.ComponentModel.Design.IDesigner", DesignerTypeName="C1.C1Excel.Design.C1XLBookDesigner, C1.C1Excel.4.Design, Version=4.1.20133.280")]
[System.ComponentModel.LicenseProviderAttribute(System.ComponentModel.LicenseProvider)]
[System.Drawing.ToolboxBitmapAttribute()]
public class C1XLBook : System.ComponentModel.Component

```

Inheritance Hierarchy

```

System.Object
    System.MarshalByRefObject
        System.ComponentModel.Component
            C1.C1Excel.C1XLBook

```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Members](#)


[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#) [Events](#)












The following tables list the members exposed by [C1XLBook](#).


Public Constructors

	Name	Description
	C1XLBook Constructor	Creates a new instance of the C1XLBook class. The new workbook contains a single empty XLSheet called "Sheet1".

[Top](#)












Public Properties

	Name	Description
	Author	Gets or sets the name of the person, company, or application that created this C1XLBook .
	CalculationMode	Gets or sets the formula calculation mode.
	CompatibilityMode	Gets or sets a value determining the limits on sheet size and the number of styles allowed per workbook.
	Container	(Inherited from System.ComponentModel.Component)
	DefaultFont	Gets or sets the default font object for the C1XLBook .
	IsLoading	Determines whether the workbook is currently loading.
	KeepFormulas	Specifies whether the component should store formulas read from XLS/XLSX files and write them back when saving the file.
	NamedRanges	Gets the collection of XLNamedRange objects for the current workbook.
	OpaqueCopy	Gets or sets a value specifying whether the component should copy non-main BIFF records as opaque when loading and saving XLS files.
	Palette	Gets or sets color palette of this workbook.
	Sheets	Gets a collection of XLSheet objects that represent the worksheets in the

		C1XLBook .
	Site	(Inherited from System.ComponentModel.Component)


[Top](#)

Public Methods

	Name	Description
	Clear	Clears the C1XLBook , restoring the initial state with a single XLSheet called "Sheet1".
	Clone	Creates a copy of this C1Excel book.
	CreateObjRef	(Inherited from System.MarshalByRefObject)
	Dispose	(Inherited from System.ComponentModel.Component)
	GetLifetimeService	(Inherited from System.MarshalByRefObject)
	InitializeLifetimeService	(Inherited from System.MarshalByRefObject)
	Load	Overloaded. Loads an Excel worksheet from a file.
 S	PixelsToTwips	Converts a pixel measurement into twips (1/20th of a point).
	Save	Overloaded. Saves the worksheet to a file.
	ToString	(Inherited from System.ComponentModel.Component)
 S	TwipsToPixels	Converts a twip measurement into screen pixels.

[Top](#)

Public Events

	Name	Description
	Disposed	(Inherited from System.ComponentModel.Component)

[Top](#)

See Also

Reference

[C1XLBook Class](#)

[C1.C1Excel Namespace](#)

C1XLBook Constructor

Creates a new instance of the [C1XLBook](#) class. The new workbook contains a single empty [XLSheet](#) called "Sheet1".

Syntax

Visual Basic (Declaration)	
Public Function New()	
C#	
public C1XLBook()	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

Methods

For a list of all members of this type, see [C1XLBook members](#).

Public Methods

	Name	Description
≡	Clear	Clears the C1XLBook , restoring the initial state with a single XLSheet called "Sheet1".
≡	Clone	Creates a copy of this C1Excel book.
≡	CreateObjRef	(Inherited from System.MarshalByRefObject)
≡	Dispose	(Inherited from System.ComponentModel.Component)
≡	GetLifetimeService	(Inherited from System.MarshalByRefObject)
≡	InitializeLifetimeService	(Inherited from System.MarshalByRefObject)
≡	Load	Overloaded. Loads an Excel worksheet from a file.
≡ S	PixelsToTwips	Converts a pixel measurement into twips (1/20th of a point).
≡	Save	Overloaded. Saves the worksheet to a file.
≡	ToString	(Inherited from System.ComponentModel.Component)
≡ S	TwipsToPixels	Converts a twip measurement into screen pixels.

[Top](#)

See Also

Reference

[C1XLBook Class](#)

[C1.C1Excel Namespace](#)

Clear Method

Clears the [C1XLBook](#), restoring the initial state with a single [XLSheet](#) called "Sheet1".

Syntax

Visual Basic (Declaration)	
Public Sub Clear()	
C#	
public void Clear()	

Remarks

The [Clear](#) method restores the [C1XLBook](#) object to its initial state, with a single empty sheet called "Sheet1" and the [DefaultFont](#) set to 10pt Arial.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

Clone Method

Creates a copy of this C1Excel book.

Syntax

Visual Basic (Declaration)	
Public Function Clone() As C1XLBook	
C#	
public C1XLBook Clone()	

Return Value

A new [C1XLBook](#) object with the same contents and formatting as this book.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

Load Method

Loads an Excel worksheet from a file.

Overload List

Overload	Description
Load(String)	Loads an Excel worksheet from a file.
Load(String,Boolean)	Loads an Excel worksheet from a file.
Load(String,FileFormat)	Loads an Excel worksheet from a file.
Load(String,FileFormat,Boolean)	Loads an Excel worksheet from a file.
Load(Stream)	Loads the worksheet from a stream.
Load(Stream,Boolean)	Loads the worksheet from a stream.
Load(Stream,FileFormat)	Loads the worksheet from a stream.
Load(Stream,FileFormat,Boolean)	Loads the worksheet from a stream.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

Load(String) Method

Name of the file that contains the worksheet.

Loads an Excel worksheet from a file.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Sub Load(_ ByVal fileName As System.String _)</pre>	
C#	
<pre>public void Load(System.string fileName)</pre>	

Parameters

fileName

Name of the file that contains the worksheet.

Remarks

Component One Excel infers the file format automatically based on the file name extension. "XLSX" and "ZIP" files are loaded as OpenXml; all others are loaded as Biff8 files ("xls").

If the file doesn't exist, is locked, or is not a valid Excel file, an exception is thrown.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

[Overload List](#)

Load(String, Boolean) Method

Name of the file that contains the worksheet.

True to load data into the sheets; False to read the sheet names only.

Loads an Excel worksheet from a file.

Syntax

Visual Basic (Declaration)

```
Public Overloads Sub Load( _  
    ByVal fileName As System.String, _  
    ByVal fillSheets As System.Boolean _  
)
```

C#

```
public void Load(  
    System.string fileName,  
    System.bool fillSheets  
)
```

Parameters

fileName

Name of the file that contains the worksheet.

fillSheets

True to load data into the sheets; False to read the sheet names only.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)
[C1XLBook Members](#)
[Overload List](#)

Load(String,FileFormat) Method

Name of the file that contains the worksheet.

FileFormat value that specifies the file format.

Loads an Excel worksheet from a file.

Syntax

Visual Basic (Declaration)

```
Public Overloads Sub Load( _  
    ByVal fileName As System.String, _  
    ByVal format As FileFormat _  
)
```

C#

```
public void Load(  
    System.string fileName,  
    FileFormat format  
)
```

Parameters

fileName

Name of the file that contains the worksheet.

format

FileFormat value that specifies the file format.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

[Overload List](#)

Load(String,FileFormat,Boolean) Method

Name of the file that contains the worksheet.

FileFormat value that specifies the file format.

True to load data into the sheets; False to read the sheet names only.

Loads an Excel worksheet from a file.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Sub Load(_ ByVal fileName As System.String, _ ByVal format As FileFormat, _ ByVal fillSheets As System.Boolean _)</pre>	
C#	
<pre>public void Load(System.string fileName, FileFormat format, System.bool fillSheets)</pre>	

Parameters

fileName

Name of the file that contains the worksheet.

format

FileFormat value that specifies the file format.

fillSheets

True to load data into the sheets; False to read the sheet names only.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

[Overload List](#)

Load(Stream) Method

System.IO.Stream that contains the worksheet.

Loads the worksheet from a stream.

Syntax

Visual Basic (Declaration)

```
Public Overloads Sub Load( _  
    ByVal stream As System.IO.Stream _  
)
```

C#

```
public void Load(  
    System.IO.Stream stream  
)
```

Parameters

stream

System.IO.Stream that contains the worksheet.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

[Overload List](#)

Load(Stream, Boolean) Method

System.IO.Stream that contains the worksheet.

True to load data into the sheets; False to read the sheet names only.

Loads the worksheet from a stream.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Sub Load(_ ByVal stream As System.IO.Stream, _ ByVal fillSheets As System.Boolean _)</pre>	
C#	
<pre>public void Load(System.IO.Stream stream, System.bool fillSheets)</pre>	

Parameters

stream

System.IO.Stream that contains the worksheet.

fillSheets

True to load data into the sheets; False to read the sheet names only.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)
[C1XLBook Members](#)
[Overload List](#)

Load(Stream,FileFormat) Method

System.IO.Stream that contains the worksheet.

FileFormat value that specifies the file format.

Loads the worksheet from a stream.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Sub Load(_ ByVal stream As System.IO.Stream, _ ByVal format As FileFormat _)</pre>	
C#	
<pre>public void Load(System.IO.Stream stream, FileFormat format)</pre>	

Parameters

stream

System.IO.Stream that contains the worksheet.

format

FileFormat value that specifies the file format.

Remarks

Loading the worksheets without their data is much faster than loading the entire workbook. This is useful in situations where you want to examine the contents of the file (for example, to ensure that you will not overwrite an existing sheet).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

[Overload List](#)

Load(Stream,FileFormat,Boolean) Method

System.IO.Stream that contains the worksheet.

FileFormat value that specifies the file format.

True to load data into the sheets; **False** to read the sheet names only.

Loads the worksheet from a stream.

Syntax

Visual Basic (Declaration)

```
Public Overloads Sub Load( _  
    ByVal stream As System.IO.Stream, _  
    ByVal format As FileFormat, _  
    ByVal fillSheets As System.Boolean _  
)
```

C#

```
public void Load(  
    System.IO.Stream stream,  
    FileFormat format,  
    System.bool fillSheets  
)
```

Parameters

stream

System.IO.Stream that contains the worksheet.

format

FileFormat value that specifies the file format.

fillSheets

True to load data into the sheets; **False** to read the sheet names only.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)
[C1XLBook Members](#)
[Overload List](#)

PixelsToTwips Method

Measurement in screen pixels.

Converts a pixel measurement into twips (1/20th of a point).

Syntax

Visual Basic (Declaration)


```
Public Shared Function PixelsToTwips( _  
    ByVal pix As System.Double _  
) As System.Integer
```

C#

```
public static System.int PixelsToTwips(  
    System.double pix  
)
```

Parameters

pix

Measurement in screen pixels.

Return Value

Measurement in twips.

Remarks

Excel stores measurements in twips (1/20th of a point), a resolution-independent unit. .NET controls, on the other hand, usually express measurements in pixels. This method provides an easy way to convert pixel measurements into twips using the current screen resolution.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

[TwipsToPixels Method](#)

Save Method

Saves the worksheet to a file.

Overload List

Overload	Description
Save(String)	Saves the worksheet to a file.
Save(String,FileFormat)	Saves the worksheet to a file.
Save(Stream)	Saves the worksheet into a stream.
Save(Stream,FileFormat)	Saves the worksheet into a stream.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

Save(String) Method

Name of the file to save.

Saves the worksheet to a file.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Sub Save(_ ByVal fileName As System.String _)</pre>	
C#	
<pre>public void Save(System.string fileName)</pre>	

Parameters

fileName

Name of the file to save.

Remarks

The format used to save the file is automatically determined by the file name extension. "Xlsx" and "zip" files are saved as OpenXml; all others are saved as Biff8 files ("xls").

If the file can't be created, an exception is thrown. This typically indicates that the file is currently open by another application (such as Microsoft Excel).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

[Overload List](#)

Save(String,FileFormat) Method

Name of the file to save.

FileFormat value that specifies the type of file to save.

Saves the worksheet to a file.

Syntax

Visual Basic (Declaration)

```
Public Overloads Sub Save( _  
    ByVal fileName As System.String, _  
    ByVal format As FileFormat _  
)
```

C#

```
public void Save(  
    System.string fileName,  
    FileFormat format  
)
```

Parameters

fileName

Name of the file to save.

format

FileFormat value that specifies the type of file to save.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

[Overload List](#)

Save(Stream) Method

[Example](#)

System.IO.Stream where the worksheet is saved.

Saves the worksheet into a stream.

Syntax

Visual Basic (Declaration)

```
Public Overloads Sub Save( _  
    ByVal stream As System.IO.Stream _  
)
```

C#

```
public void Save(  
    System.IO.Stream stream  
)
```

Parameters

stream

System.IO.Stream where the worksheet is saved.

Remarks

This method allows saving the workbook directly into streams without using temporary files. Typical uses include saving books to web page response streams or mail attachment streams.

Example

The code below saves a [C1XLBook](#) into a **System.IO.MemoryStream**, clears the book, then loads it back from the same stream.

- [C#](#)

```
// save book into new MemoryStream  
MemoryStream ms = new MemoryStream();  
_book.Save(ms);  
  
// clear book  
_book.Clear();  
  
// load it back from the MemoryStream  
ms.Position = 0;  
_book.Load(ms);
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)
[C1XLBook Members](#)
[Overload List](#)

Save(Stream,FileFormat) Method

System.IO.Stream where the worksheet is saved.

FileFormat value that specifies the format to save the worksheet in.

Saves the worksheet into a stream.

Syntax

Visual Basic (Declaration)

```
Public Overloads Sub Save( _  
    ByVal stream As System.IO.Stream, _  
    ByVal format As FileFormat _  
)
```

C#

```
public void Save(  
    System.IO.Stream stream,  
    FileFormat format  
)
```

Parameters

stream

System.IO.Stream where the worksheet is saved.

format

FileFormat value that specifies the format to save the worksheet in.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)
[C1XLBook Members](#)
[Overload List](#)

TwipsToPixels Method
Measurement in twips.

Converts a twip measurement into screen pixels.

Syntax

Visual Basic (Declaration)

```
Public Shared Function TwipsToPixels( _  
    ByVal twip As System.Double _  
) As System.Integer
```

C#

```
public static System.int TwipsToPixels(  
    System.double twip  
)
```

Parameters

twip

Measurement in twips.

Return Value

Measurement in screen pixels.

Remarks

Excel stores measurements in twips (1/20th of a point), a resolution-independent unit. .NET controls, on the other hand, usually express measurements in pixels. This method provides an easy way to convert pixel measurements into twips using the current screen resolution.

[PixelsToTwips](#)

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference










[C1XLBook Class](#)




[C1XLBook Members](#)

Properties

For a list of all members of this type, see [C1XLBook members](#).

Public Properties

	Name	Description
	Author	Gets or sets the name of the person, company, or application that created this C1XLBook .
	CalculationMode	Gets or sets the formula calculation mode.
	CompatibilityMode	Gets or sets a value determining the limits on sheet size and the number of styles allowed per workbook.
	Container	(Inherited from System.ComponentModel.Component)
	DefaultFont	Gets or sets the default font object for the C1XLBook .
	IsLoading	Determines whether the workbook is currently loading.
	KeepFormulas	Specifies whether the component should store formulas read from XLS/XLSX files and write them back when saving the file.
	NamedRanges	Gets the collection of XLNamedRange objects for the current workbook.
	OpaqueCopy	Gets or sets a value specifying whether the component should copy non-

		main BIFF records as opaque when loading and saving XLS files.
	Palette	Gets or sets color palette of this workbook.
	Sheets	Gets a collection of XLSheet objects that represent the worksheets in the C1XLBook .
	Site	(Inherited from System.ComponentModel.Component)

[Top](#)

See Also

Reference

[C1XLBook Class](#)

[C1.C1Excel Namespace](#)

Author Property

Gets or sets the name of the person, company, or application that created this [C1XLBook](#).

Syntax

Visual Basic (Declaration)

```
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializationVisibility.Hidden)>
<System.ComponentModel.BrowsableAttribute(False)>
Public Property Author As System.String
```

C#

```
[System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializationVisibility.Hidden)]
[System.ComponentModel.BrowsableAttribute(false)]
public System.string Author {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

CalculationMode Property

Gets or sets the formula calculation mode.

Syntax

Visual Basic (Declaration)	
<pre><System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializationVisibility.Hidden)> <System.ComponentModel.BrowsableAttribute(False)> Public Property CalculationMode As CalculationMode</pre>	
C#	
<pre>[System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializationVisibility.Hidden)] [System.ComponentModel.BrowsableAttribute(false)] public CalculationMode CalculationMode {get; set;}</pre>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

CompatibilityMode Property

Gets or sets a value determining the limits on sheet size and the number of styles allowed per workbook.

Syntax

Visual Basic (Declaration)

```
<C1DescriptionAttribute("Determines the limits on sheet size and the number of  
styles allowed per workbook.")>  
<System.ComponentModel.DefaultValueAttribute()>  
Public Property CompatibilityMode As CompatibilityMode
```

C#

```
[C1DescriptionAttribute("Determines the limits on sheet size and the number of  
styles allowed per workbook.")]  
[System.ComponentModel.DefaultValueAttribute()]  
public CompatibilityMode CompatibilityMode {get; set;}
```

Remarks

This property allows you to specify which version of Microsoft Excel you want your workbooks to be compatible with.

Excel2003 mode allows you to create sheets with up to 65,536 rows and 256 columns. **Excel2007** mode allows you to create sheets with up to 1,048,576 rows and 18,278 columns.

Note that the XLS file format is limited by the **Excel2003** limits. If you use the **Excel2007** mode and create large sheets, save them into **OpenXml** files instead of XLS.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

DefaultFont Property

Gets or sets the default font object for the [C1XLBook](#).

Syntax

Visual Basic (Declaration)	
<pre><System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializationVisibility.Hidden)> <System.ComponentModel.BrowsableAttribute(False)> Public Property DefaultFont As System.Drawing.Font</pre>	
C#	
<pre>[System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializationVisibility.Hidden)] [System.ComponentModel.BrowsableAttribute(false)] public System.Drawing.Font DefaultFont {get; set;}</pre>	

Remarks

You can assign any font to any cell using [XLStyle](#) objects. Cells that have no associated custom styles or have styles that do not define a custom font are displayed using the book's default font.

The [DefaultFont](#) property is initially set to 10 pt Arial.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

IsLoading Property

Determines whether the workbook is currently loading.

Syntax

Visual Basic (Declaration)	
<pre><System.ComponentModel.BrowsableAttribute(False)> <System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializ</pre>	

```
ationVisibility.Hidden)>
Public ReadOnly Property IsLoading As System.Boolean
```

C#

```
[System.ComponentModel.BrowsableAttribute(false)]
[System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializ
ationVisibility.Hidden)]
public System.bool IsLoading {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

KeepFormulas Property

Specifies whether the component should store formulas read from XLS/XLSX files and write them back when saving the file.

Syntax

Visual Basic (Declaration)

```
<System.ComponentModel.DefaultValueAttribute()>
<C1DescriptionAttribute("Specifies whether the component should preserve
formulas when loading and saving XLS/XLSX files.")>
Public Property KeepFormulas As System.Boolean
```

C#

```
[System.ComponentModel.DefaultValueAttribute()]
[C1DescriptionAttribute("Specifies whether the component should preserve
formulas when loading and saving XLS/XLSX files.")]
public System.bool KeepFormulas {get; set;}
```

Remarks

Setting this property to true allows you to load existing XLS/XLSX files, modify the values in some cells, and save the file preserving the formulas. This is the default setting.

Setting this property to false causes the component to remove the formulas in the book when it is loaded. Saving the file in this case will retain the last calculated values but will remove the formulas.

Assigning any value to a cell will clear the formula in the cell.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

NamedRanges Property

Gets the collection of [XLNamedRange](#) objects for the current workbook.

Syntax

Visual Basic (Declaration)	
<pre><System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializationVisibility.Hidden)> <System.ComponentModel.BrowsableAttribute(False)> Public ReadOnly Property NamedRanges As XLNamedRangeCollection</pre>	
C#	
<pre>[System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializationVisibility.Hidden)] [System.ComponentModel.BrowsableAttribute(false)] public XLNamedRangeCollection NamedRanges {get;}</pre>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

OpaqueCopy Property

Gets or sets a value specifying whether the component should copy non-main BIFF records as opaque when loading and saving XLS files.

Syntax

Visual Basic (Declaration)

```
<C1DescriptionAttribute("Specifies whether the component should copy non-main  
BIFF records as opaque when loading and saving XLS files.")>  
<System.ComponentModel.DefaultValueAttribute()>  
Public Property OpaqueCopy As System.Boolean
```

C#

```
[C1DescriptionAttribute("Specifies whether the component should copy non-main  
BIFF records as opaque when loading and saving XLS files.")]  
[System.ComponentModel.DefaultValueAttribute()]  
public System.bool OpaqueCopy {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

Palette Property

Gets or sets color palette of this workbook.

Syntax

Visual Basic (Declaration)

```
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializ  
ationVisibility.Hidden)>  
<System.ComponentModel.BrowsableAttribute(False)>  
Public Property Palette As System.Drawing.Color()
```

C#

```
[System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializ  
ationVisibility.Hidden)]  
[System.ComponentModel.BrowsableAttribute(false)]  
public System.Drawing.Color[] Palette {get; set;}
```

Remarks

The palette must be more 8 items (first 8 colors is standard pallete: Black, White, Red, Green, Blue, Yellow, Magenta, Cyan).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

Sheets Property

Gets a collection of [XLSheet](#) objects that represent the worksheets in the [C1XLBook](#).

Syntax

Visual Basic (Declaration)	
<pre><System.ComponentModel.BrowsableAttribute(False)> Public ReadOnly Property Sheets As XLSheetCollection</pre>	
C#	
<pre>[System.ComponentModel.BrowsableAttribute(false)] public XLSheetCollection Sheets {get;}</pre>	

Remarks

The [XLSheetCollection](#) returned has methods for counting, enumerating, adding and removing sheets from the [C1XLBook](#).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1XLBook Class](#)

[C1XLBook Members](#)

ShapeCollection

Represents a collection of [XLShape](#) objects on an [XLSheet](#).

Object Model

ShapeCollection

Syntax

Visual Basic (Declaration)	
<pre><System.Reflection.DefaultMemberAttribute("Item")> Public Class ShapeCollection</pre>	

C#

```
[System.Reflection.DefaultMemberAttribute("Item")]  
public class ShapeCollection
```

Inheritance Hierarchy

System.Object

C1.C1Excel.ShapeCollection

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents a collection of [XLShape](#) objects on an [XLSheet](#).

Object Model

ShapeCollection

Syntax

Visual Basic (Declaration)

```
<System.Reflection.DefaultMemberAttribute("Item")>  
Public Class ShapeCollection
```

C#

```
[System.Reflection.DefaultMemberAttribute("Item")]  
public class ShapeCollection
```

Inheritance Hierarchy

System.Object

C1.C1Excel.ShapeCollection

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Members](#)




[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)



The following tables list the members exposed by [ShapeCollection](#).




Public Properties

	Name	Description
	Count	Gets count items in collection.
	Item	Gets the XLShape object at the specified position in the collection.
	Sheet	Gets the Sheet object that owns the collection.

[Top](#)

Public Methods

	Name	Description
	Add	Appends an XLShape object to the collection.
	Contains	Overloaded. Checks whether the collection contains a specific XLShape object.

 IndexOf	Overloaded. Gets the position of an XLShape object in the collection.
 Insert	Inserts an XLShape object at a specific position in the collection.
 Remove	Overloaded. Removes an XLShape object from the collection.

[Top](#)

See Also






Reference

[ShapeCollection Class](#)
[C1.C1Excel Namespace](#)

Methods

For a list of all members of this type, see [ShapeCollection members](#).

Public Methods

	Name	Description
 Add		Appends an XLShape object to the collection.
 Contains		Overloaded. Checks whether the collection contains a specific XLShape object.
 IndexOf		Overloaded. Gets the position of an XLShape object in the collection.
 Insert		Inserts an XLShape object at a specific position in the collection.
 Remove		Overloaded. Removes an XLShape object from the collection.

[Top](#)

See Also

Reference

[ShapeCollection Class](#)
[C1.C1Excel Namespace](#)

Add Method

[XLShape](#) object to add to the collection.

Appends an [XLShape](#) object to the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Function Add(_ ByVal shape As XLShape _) As XLShape</pre>	
C#	
<pre>public XLShape Add(XLShape shape)</pre>	

Parameters

shape

[XLShape](#) object to add to the collection.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added to the collection (usually because it overlaps another cell range already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Class](#)

[ShapeCollection Members](#)

Contains Method

Checks whether the collection contains a specific [XLShape](#) object.

Overload List

Overload	Description
Contains(XLShape)	Checks whether the collection contains a specific XLShape object.
Contains(Int32)	Checks whether the collection contains a specific XLShape object.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Class](#)

[ShapeCollection Members](#)

Contains(XLShape) Method

[XLShape](#) object to look for.

Checks whether the collection contains a specific [XLShape](#) object.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function Contains(_ ByVal shape As XLShape _) As System.Boolean</pre>	
C#	
<pre>public System.bool Contains(XLShape shape)</pre>	

Parameters

shape

[XLShape](#) object to look for.

Return Value

True if the collection contains the range, False otherwise.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Class](#)

[ShapeCollection Members](#)

[Overload List](#)

Contains(Int32) Method

The identifier of the object to look for.

Checks whether the collection contains a specific [XLShape](#) object.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function Contains(_ ByVal id As System.Integer _) As System.Boolean</pre>	
C#	
<pre>public System.bool Contains(System.int id)</pre>	

Parameters

id

The identifier of the object to look for.

Return Value

True if the collection contains the range, False otherwise.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Class](#)
[ShapeCollection Members](#)
[Overload List](#)

IndexOf Method

Gets the position of an [XLShape](#) object in the collection.

Overload List

Overload	Description
IndexOf(XLShape)	Gets the position of an XLShape object in the collection.
IndexOf(Int32)	Gets the position of an XLShape object in the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Class](#)
[ShapeCollection Members](#)

IndexOf(XLShape) Method

[XLShape](#) object to look for.

Gets the position of an [XLShape](#) object in the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function IndexOf(_ ByVal shape As XLShape _) As System.Integer</pre>	
C#	
<pre>public System.int IndexOf(XLShape shape)</pre>	

Parameters

shape

[XLShape](#) object to look for.

Return Value

The position of the object in the collection, or -1 if the object is not a member of the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Class](#)

[ShapeCollection Members](#)

[Overload List](#)

IndexOf(Int32) Method

The identifier of the object to look for.

Gets the position of an [XLShape](#) object in the collection.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function IndexOf( _  
    ByVal id As System.Integer _  
) As System.Integer
```

C#

```
public System.int IndexOf(  
    System.int id  
)
```

Parameters

id

The identifier of the object to look for.

Return Value

The position of the object in the collection, or -1 if the object is not a member of the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Class](#)

[ShapeCollection Members](#)

[Overload List](#)

Insert Method

Position where the object will be inserted.

Object to insert in the collection.

Inserts an [XLShape](#) object at a specific position in the collection.

Syntax

Visual Basic (Declaration)

```
Public Overridable Function Insert( _  
    ByVal index As System.Integer, _  
    ByVal shape As XLShape _  
) As XLShape
```

C#

```
public virtual XLShape Insert(  
    System.int index,  
    XLShape shape  
)
```

Parameters

index

Position where the object will be inserted.

shape

Object to insert in the collection.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added to the collection (usually because it overlaps another cell range already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Class](#)

[ShapeCollection Members](#)

Remove Method

Removes an [XLShape](#) object from the collection.

Overload List

Overload	Description
Remove(XLShape)	Removes an XLShape object from the collection.
Remove(Int32)	Removes an XLShape object from the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Class](#)

[ShapeCollection Members](#)

Remove(XLShape) Method

[XLShape](#) object to remove from the collection.

Removes an [XLShape](#) object from the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function Remove(_ ByVal shape As XLShape _) As System.Boolean</pre>	
C#	
<pre>public System.bool Remove(XLShape shape)</pre>	

Parameters

shape

[XLShape](#) object to remove from the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Class](#)

[ShapeCollection Members](#)

[Overload List](#)

Remove(Int32) Method

The identifier of the object to remove from the collection.

Removes an [XLShape](#) object from the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function Remove(_ ByVal id As System.Integer _) As System.Boolean</pre>	
C#	
<pre>public System.bool Remove(System.int id)</pre>	

Parameters

id

The identifier of the object to remove from the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also




Reference

[ShapeCollection Class](#)
[ShapeCollection Members](#)
[Overload List](#)

Properties

For a list of all members of this type, see [ShapeCollection members](#).

Public Properties

	Name	Description
	Count	Gets count items in collection.
	Item	Gets the XLShape object at the specified position in the collection.
	Sheet	Gets the Sheet object that owns the collection.

[Top](#)

See Also

Reference

[ShapeCollection Class](#)
[C1.C1Excel Namespace](#)

Count Property
Gets count items in collection.

Syntax

Visual Basic (Declaration)	
<pre>Public ReadOnly Property Count As System.Integer</pre>	

C#

```
public System.int Count {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Class](#)

[ShapeCollection Members](#)

Item Property

Gets the [XLShape](#) object at the specified position in the collection.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Default Property Item( _  
    ByVal index As System.Integer _  
) As XLShape
```

C#

```
public XLShape this[  
    System.int index  
]; {get;}
```

Parameters

index

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Class](#)

[ShapeCollection Members](#)

Sheet Property

Gets the [Sheet](#) object that owns the collection.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property Sheet As XLSheet
```

C#

```
public XLSheet Sheet {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[ShapeCollection Class](#)

[ShapeCollection Members](#)

Strings

Static class containing UI strings used by the designer.

Object Model

Strings

Syntax

Visual Basic (Declaration)

Public MustInherit NotInheritable Class Strings

C#

public static class Strings

Inheritance Hierarchy

System.Object

C1.C1Excel.Strings

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Members](#)

[C1.C1Excel Namespace](#)

Overview

Static class containing UI strings used by the designer.

Object Model

Strings

Syntax

Visual Basic (Declaration)

Public MustInherit NotInheritable Class Strings

C#

public static class Strings

Inheritance Hierarchy

System.Object

C1.C1Excel.Strings

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Members](#)



[C1.C1Excel Namespace](#)



















Members












[Properties](#)

The following tables list the members exposed by [Strings](#).

Public Properties

	Name	Description
 S	BadFormatRunDataLenFmt	
 S	BadOpenXmlFile	
 S	BadParamInSheetCtor	
 S	BadParamInStyleCtor	
 S	CannotCreateStorageFile	
 S	CannotCreateStorageStream	
 S	CannotCreateStream	
 S	CannotOpenStorageFile	

 S	CannotOpenStream	
 S	CannotSaveBookTooManyCols	
 S	CannotSaveBookTooManyRows	
 S	CannotSaveBookTooManyStyles	
 S	CannotSaveEmptyBook	
 S	CannotSaveWithoutSheets	
 S	CannotSeekToEndOfStream	
 S	CannotSetStreamLength	
 S	ColumnAlreadyInSheet	
 S	ColumnFromOtherSheet	
 S	DuplicateSheetName	
 S	FilePasswordProtected	
 S	InvalidFileFormat	
 S	InvalidSheetName	
 S	MaxNoOfSheetsExceeded	
 S	NamedRangeAlreadyInBook	
 S	NegativeStartIndexOfCellRange	
 S	ResourceManager	

 S	RowAlreadyInSheet	
 S	RowFromOtherSheet	
 S	ShapeAlreadyInSheet	
 S	SheetAlreadyInBook	
 S	SheetFromOtherBook	
 S	SheetNameInUse	
 S	SheetNotFoundFmt	
 S	TooManyColsFmt	
 S	TooManyRowsFmt	
 S	UICulture	
 S	WorkbookMustBeLoaded	

[Top](#)

See Also

Reference

[Strings Class](#)



















[C1.C1Excel Namespace](#)



















Properties

For a list of all members of this type, see [Strings members](#).

Public Properties

	Name	Description
--	------	-------------

 S	BadFormatRunDataLenFmt	
 S	BadOpenXmlFile	
 S	BadParamInSheetCtor	
 S	BadParamInStyleCtor	
 S	CannotCreateStorageFile	
 S	CannotCreateStorageStream	
 S	CannotCreateStream	
 S	CannotOpenStorageFile	
 S	CannotOpenStream	
 S	CannotSaveBookTooManyCols	
 S	CannotSaveBookTooManyRows	
 S	CannotSaveBookTooManyStyles	
 S	CannotSaveEmptyBook	
 S	CannotSaveWithoutSheets	
 S	CannotSeekToEndOfStream	
 S	CannotSetStreamLength	
 S	ColumnAlreadyInSheet	
 S	ColumnFromOtherSheet	

 S	DuplicateSheetName	
 S	FilePasswordProtected	
 S	InvalidFileFormat	
 S	InvalidSheetName	
 S	MaxNoOfSheetsExceeded	
 S	NamedRangeAlreadyInBook	
 S	NegativeStartIndexOfCellRange	
 S	ResourceManager	
 S	RowAlreadyInSheet	
 S	RowFromOtherSheet	
 S	ShapeAlreadyInSheet	
 S	SheetAlreadyInBook	
 S	SheetFromOtherBook	
 S	SheetNameInUse	
 S	SheetNotFoundFmt	
 S	TooManyColsFmt	
 S	TooManyRowsFmt	
 S	UICulture	

 S	WorkbookMustBeLoaded	
--	----------------------	--

[Top](#)

See Also

Reference

[Strings Class](#)

[C1.C1Excel Namespace](#)

BadFormatRunDataLenFmt Property

Syntax

Visual Basic (Declaration)	
<code>Public Shared ReadOnly Property BadFormatRunDataLenFmt As System.String</code>	
C#	
<code>public static System.string BadFormatRunDataLenFmt {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

BadOpenXmlFile Property

Syntax

Visual Basic (Declaration)	
<code>Public Shared ReadOnly Property BadOpenXmlFile As System.String</code>	
C#	

```
public static System.string BadOpenXmlFile {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

BadParamInSheetCtor Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property BadParamInSheetCtor As System.String
```

C#

```
public static System.string BadParamInSheetCtor {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

BadParamInStyleCtor Property

Syntax

Visual Basic (Declaration)

Public Shared ReadOnly Property BadParamInStyleCtor As System.String
--

C#

public static System.string BadParamInStyleCtor {get;}
--

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

[CannotCreateStorageFile Property](#)

Syntax

Visual Basic (Declaration)

Public Shared ReadOnly Property CannotCreateStorageFile As System.String
--

C#

public static System.string CannotCreateStorageFile {get;}
--

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

CannotCreateStorageStream Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property CannotCreateStorageStream As System.String
```

C#

```
public static System.string CannotCreateStorageStream {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

CannotCreateStream Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property CannotCreateStream As System.String
```

C#

```
public static System.string CannotCreateStream {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)
[Strings Members](#)

CannotOpenStorageFile Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property CannotOpenStorageFile As System.String
```

C#

```
public static System.string CannotOpenStorageFile {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)
[Strings Members](#)

CannotOpenStream Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property CannotOpenStream As System.String
```

C#

```
public static System.string CannotOpenStream {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

[CannotSaveBookTooManyCols Property](#)

Syntax

Visual Basic (Declaration)	
<code>Public Shared ReadOnly Property CannotSaveBookTooManyCols As System.String</code>	
C#	
<code>public static System.string CannotSaveBookTooManyCols {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

[CannotSaveBookTooManyRows Property](#)

Syntax

Visual Basic (Declaration)	
<code>Public Shared ReadOnly Property CannotSaveBookTooManyRows As System.String</code>	
C#	
<code>public static System.string CannotSaveBookTooManyRows {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

CannotSaveBookTooManyStyles Property

Syntax

Visual Basic (Declaration)	
Public Shared ReadOnly Property CannotSaveBookTooManyStyles As System.String	
C#	
public static System.string CannotSaveBookTooManyStyles { get ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

CannotSaveEmptyBook Property

Syntax

Visual Basic (Declaration)	
Public Shared ReadOnly Property CannotSaveEmptyBook As System.String	
C#	

```
public static System.string CannotSaveEmptyBook {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

CannotSaveWithoutSheets Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property CannotSaveWithoutSheets As System.String
```

C#

```
public static System.string CannotSaveWithoutSheets {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

CannotSeekToEndOfStream Property

Syntax

Visual Basic (Declaration)

Public Shared ReadOnly Property CannotSeekToEndOfStream As System.String
--

C#

public static System.string CannotSeekToEndOfStream {get;}
--

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

CannotSetStreamLength Property

Syntax

Visual Basic (Declaration)

Public Shared ReadOnly Property CannotSetStreamLength As System.String
--

C#

public static System.string CannotSetStreamLength {get;}
--

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

ColumnAlreadyInSheet Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property ColumnAlreadyInSheet As System.String
```

C#

```
public static System.string ColumnAlreadyInSheet {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

ColumnFromOtherSheet Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property ColumnFromOtherSheet As System.String
```

C#

```
public static System.string ColumnFromOtherSheet {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)
[Strings Members](#)

DuplicateSheetName Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property DuplicateSheetName As System.String
```

C#

```
public static System.string DuplicateSheetName {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)
[Strings Members](#)

FilePasswordProtected Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property FilePasswordProtected As System.String
```

C#

```
public static System.string FilePasswordProtected {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

[InvalidFileFormat Property](#)

Syntax

Visual Basic (Declaration)	
<code>Public Shared ReadOnly Property InvalidFileFormat As System.String</code>	
C#	
<code>public static System.string InvalidFileFormat {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

[InvalidSheetName Property](#)

Syntax

Visual Basic (Declaration)	
<code>Public Shared ReadOnly Property InvalidSheetName As System.String</code>	
C#	
<code>public static System.string InvalidSheetName {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

MaxNoOfSheetsExceeded Property

Syntax

Visual Basic (Declaration)	
Public Shared ReadOnly Property MaxNoOfSheetsExceeded As System.String	
C#	
public static System.string MaxNoOfSheetsExceeded { get ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

NamedRangeAlreadyInBook Property

Syntax

Visual Basic (Declaration)	
Public Shared ReadOnly Property NamedRangeAlreadyInBook As System.String	
C#	

```
public static System.string NamedRangeAlreadyInBook {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

NegativeStartIndexOfCellRange Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property NegativeStartIndexOfCellRange As System.String
```

C#

```
public static System.string NegativeStartIndexOfCellRange {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

ResourceManager Property

Syntax

Visual Basic (Declaration)

<code>Public Shared Property ResourceManager As System.Resources.ResourceManager</code>	
C#	
<code>public static System.Resources.ResourceManager ResourceManager {get; set;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

[RowAlreadyInSheet Property](#)

Syntax

Visual Basic (Declaration)	
<code>Public Shared ReadOnly Property RowAlreadyInSheet As System.String</code>	
C#	
<code>public static System.string RowAlreadyInSheet {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

RowFromOtherSheet Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property RowFromOtherSheet As System.String
```

C#

```
public static System.string RowFromOtherSheet {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

ShapeAlreadyInSheet Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property ShapeAlreadyInSheet As System.String
```

C#

```
public static System.string ShapeAlreadyInSheet {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)
[Strings Members](#)

SheetAlreadyInBook Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property SheetAlreadyInBook As System.String
```

C#

```
public static System.string SheetAlreadyInBook {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)
[Strings Members](#)

SheetFromOtherBook Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property SheetFromOtherBook As System.String
```

C#

```
public static System.string SheetFromOtherBook {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

SheetNameInUse Property

Syntax

Visual Basic (Declaration)	
<code>Public Shared ReadOnly Property SheetNameInUse As System.String</code>	
C#	
<code>public static System.string SheetNameInUse {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

SheetNotFoundFmt Property

Syntax

Visual Basic (Declaration)	
<code>Public Shared ReadOnly Property SheetNotFoundFmt As System.String</code>	
C#	
<code>public static System.string SheetNotFoundFmt {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

TooManyColsFmt Property

Syntax

Visual Basic (Declaration)	
<code>Public Shared ReadOnly Property TooManyColsFmt As System.String</code>	
C#	
<code>public static System.string TooManyColsFmt {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

TooManyRowsFmt Property

Syntax

Visual Basic (Declaration)	
<code>Public Shared ReadOnly Property TooManyRowsFmt As System.String</code>	
C#	

```
public static System.string TooManyRowsFmt {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

UICulture Property

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly Property UICulture As System.Globalization.CultureInfo
```

C#

```
public static System.Globalization.CultureInfo UICulture {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

WorkbookMustBeLoaded Property

Syntax

Visual Basic (Declaration)

Public Shared ReadOnly Property WorkbookMustBeLoaded **As** System.String

C#

```
public static System.string WorkbookMustBeLoaded {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[Strings Class](#)

[Strings Members](#)

XLCell

[Example](#)

Represents individual cells in an [XLSheet](#) and provides properties for getting and setting the cell [Value](#), [Style](#), and [Hyperlink](#).

Object Model

XLCell

Syntax

Visual Basic (Declaration)

```
Public Class XLCell
```

C#

```
public class XLCell
```

Remarks

To create cells, use the [XLSheet](#) indexer (Item property). If the cell already exists, the reference will be returned as usual. If not, the sheet will create the cell (as well as rows and columns if necessary) and will return a reference to the new cell.

Because it creates cells automatically, the indexer is especially useful when creating and populating sheets.

Example

For example, the code below creates a new [C1XLBook](#), then populates the first sheet with a 10 by 10 multiplication table: Note how the code simply accesses the cells using the indexer. There's no need to create any rows, columns, or cells. The indexer takes care of all that automatically.

- [C#](#)

```
C1XLBook book = new C1XLBook();
XLSheet sheet = book.Sheets[0];
for (int r = 0; r < 10; r++)
{
    for (int c = 0; c < 10; c++)
    {
        XLCell cell = sheet[r, c];
        cell.Value = (r+1) * (c+1);
    }
}
book.Save(@"c:\temp\test.xls");
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLCell

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCell Members](#)

[C1.C1Excel Namespace](#)

Overview

[Example](#)

Represents individual cells in an [XLSheet](#) and provides properties for getting and setting the cell [Value](#), [Style](#), and [Hyperlink](#).

Object Model

XLCell

Syntax

Visual Basic (Declaration)	
<code>Public Class XLCell</code>	
C#	
<code>public class XLCell</code>	

Remarks

To create cells, use the [XLSheet](#) indexer (Item property). If the cell already exists, the reference will be returned as usual. If not, the sheet will create the cell (as well as rows and columns if necessary) and will return a reference to the new cell.

Because it creates cells automatically, the indexer is especially useful when creating and populating sheets.

Example

For example, the code below creates a new [C1XLBook](#), then populates the first sheet with a 10 by 10 multiplication table: Note how the code simply accesses the cells using the indexer. There's no need to create any rows, columns, or cells. The indexer takes care of all that automatically.

- [C#](#)

```
C1XLBook book = new C1XLBook();
XLSheet sheet = book.Sheets[0];
for (int r = 0; r < 10; r++)
{
    for (int c = 0; c < 10; c++)
    {
        XLCell cell = sheet[r, c];
        cell.Value = (r+1) * (c+1);
    }
}
book.Save(@"c:\temp\test.xls");
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLCell

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCell Members](#)






[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)

The following tables list the members exposed by [XLCell](#).



Public Properties

	Name	Description
	Formula	Gets or sets a string that specifies a formula of the cell.
	Hyperlink	Gets or sets a string that specifies an action to take when the cell is clicked.
	Style	Gets or sets the XLStyle object associated with the cell.
	Text	Gets the string representation of the current cell value.
	Value	Gets or sets the value stored in the cell.

[Top](#)

Public Methods

	Name	Description
--	------	-------------

	Clone	Creates a copy of the current cell, including the value.
	SetValue	Sets the Value and Style properties of a cell.

[Top](#)

See Also

Reference



[XLCell Class](#)

[C1.C1Excel Namespace](#)

Methods

For a list of all members of this type, see [XLCell members](#).

Public Methods

	Name	Description
	Clone	Creates a copy of the current cell, including the value.
	SetValue	Sets the Value and Style properties of a cell.

[Top](#)

See Also

Reference

[XLCell Class](#)

[C1.C1Excel Namespace](#)

Clone Method

Creates a copy of the current cell, including the value.

Syntax

Visual Basic (Declaration)	
Public Function Clone () As XLCell	

C#

```
public XlCell Clone()
```

Return Value

A new [XlCell](#) object that is a copy of the current instance.

Remarks

The formula of the cell is cloned only as part of a row, a worksheet or a workbook.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XlCell Class](#)

[XlCell Members](#)

SetValue Method

New cell value.

New cell style.

Sets the [Value](#) and [Style](#) properties of a cell.

Syntax

Visual Basic (Declaration)

```
Public Sub SetValue( _  
    ByVal value As System.Object, _  
    ByVal style As XlStyle _  
)
```

C#

```
public void SetValue(
```



```
System.object value,  
XLStyle style  
)
```

Parameters

value

New cell value.

style

New cell style.

Remarks

This method allows you to set the Value and Style properties of a cell simultaneously. This can make your code more compact and easier to maintain. For example: `// set cell value and style (short version) sheet[0,0].SetValue("Hello", styleBold); // set cell value and style (longer version) sheet[0,0].Value = "Hello"; sheet[0,0].Style = styleBold;`

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference


[XLCell Class](#)





[XLCell Members](#)

Properties

For a list of all members of this type, see [XLCell members](#).

Public Properties

	Name	Description
	Formula	Gets or sets a string that specifies a formula of the cell.

	Hyperlink	Gets or sets a string that specifies an action to take when the cell is clicked.
	Style	Gets or sets the XLStyle object associated with the cell.
	Text	Gets the string representation of the current cell value.
	Value	Gets or sets the value stored in the cell.

[Top](#)

See Also

Reference

[XLCell Class](#)

[C1.C1Excel Namespace](#)

Formula Property

Gets or sets a string that specifies a formula of the cell.

Syntax

Visual Basic (Declaration)	
Public Property Formula As System.String	
C#	
public System.string Formula { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCell Class](#)

[XLCell Members](#)

Hyperlink Property

Gets or sets a string that specifies an action to take when the cell is clicked.

Syntax

Visual Basic (Declaration)

```
Public Property Hyperlink As System.String
```

C#

```
public System.string Hyperlink {get; set;}
```

Remarks

Hyperlinks may contain URLs that when clicked open a browser window and navigate to the specified site (for example, "http://www.componentone.com"). They may also contain references to files that are launched by the application associated with the file type (for example, "readme.doc"). Finally, hyperlinks can be used to send e-mails (for example, "mailto:joe@foo.com").

Each cell may contain a hyperlink and a value. However, if you assign a hyperlink to a cell that has no value (`Value == null`), then the hyperlink text is automatically assigned to the cell value as well.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCeIl Class](#)

[XLCeIl Members](#)

Style Property

Gets or sets the [XLStyle](#) object associated with the cell.

Syntax

Visual Basic (Declaration)

```
Public Property Style As XLStyle
```

C#	
----	--

<code>public XLStyle Style {get; set;}</code>	
---	--

Remarks

The appearance of each cell is defined by one or more XLStyle objects.

When displaying a cell, Excel combines the row, column, and cell styles and merges the style elements defined in each one in order to determine how the cell should be displayed.

The precedence of the styles is: (1) cell, (2) row, (3) column, (4) default style. For example, if a cell style defines the font and background color, those will be applied regardless of the settings in the row and column styles. If the row style defines an alignment, that will be applied regardless of the column style, and so on.

The cell style may be null, in which case the cell is displayed using the other styles available or the default book style if no others are available.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCell Class](#)

[XLCell Members](#)

Text Property

Gets the string representation of the current cell value.

Syntax

Visual Basic (Declaration)	
----------------------------	--

<code>Public ReadOnly Property Text As System.String</code>	
---	--

C#	
----	--

<code>public System.string Text {get;}</code>	
---	--

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCeIl Class](#)

[XLCeIl Members](#)

Value Property

Gets or sets the value stored in the cell.

Syntax

Visual Basic (Declaration)

```
Public Property Value As System.Object
```

C#

```
public System.object Value {get; set;}
```

Remarks

The value may contain strings, numeric, Boolean, DateTime, or null objects. Other types of objects cannot be saved in Excel files.

DateTime values are internally converted into doubles, and stored in the sheet as such. The only way to tell the difference between a DateTime value and a double in Excel is by way of the format associated with the cell ([XLStyle.Format](#) property).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCeIl Class](#)

[XLCeIl Members](#)

XLCeIlRange

Represents a range of [XLCeIl](#) objects in an [XLSheet](#).

Object Model

[XLCeIlRange](#)

Syntax

Visual Basic (Declaration)

```
Public Class XLCeIlRange
```

C#

```
public class XLCeIlRange
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLCeIlRange

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCeIlRange Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents a range of [XLCeIl](#) objects in an [XLSheet](#).

Object Model

Syntax

Visual Basic (Declaration)	
Public Class XLCellRange	
C#	
public class XLCellRange	

Inheritance Hierarchy

System.Object

C1.C1Excel.XLCellRange

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Members](#)


[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)

















The following tables list the members exposed by [XLCellRange](#).




Public Constructors

	Name	Description
	XLCellRange Constructor	Overloaded.

[Top](#)





Public Properties

	Name	Description
	Book	Gets a reference to the parent C1XLBook object.
	ColumnCount	Gets the number of columns in the range.
	ColumnFrom	Gets the index of the left column in the range.
	ColumnFromRef	Gets the reference type of the left column in the range.
	ColumnTo	Gets or sets the index of the last column in the range.
	ColumnToRef	Gets the reference type of the right column in the range.
	IsEmpty	Determines whether the range is empty.
	RangeType	Gets or sets a type (default, indirect or offset) for this cell range.
	RowCount	Gets the number of rows in the range.
	RowFrom	Gets the index of the top row in the range.
	RowFromRef	Gets the reference type of the top row in the range.
	RowTo	Gets or sets the index of the last row in the range.
	RowToRef	Gets the reference type of the bottom row in the range.
	SheetCount	Gets the number of sheets in the range.
	SheetFrom	Gets the index of the first sheet in the range.
	Sheets	Gets an array of XLSheet objects.

	SheetTo	Gets or sets the index of the last sheet in the range.
	Style	Gets or sets the XLStyle object associated with this range of cells.
	Value	Gets or sets the complex value associated with this range of cells.

[Top](#)

Public Methods

	Name	Description
	Clone	Creates a copy of this cell range.
	Contains	Overloaded. Determines whether the range contains a specific cell.
	Intersects	Determines whether the range intersects another range.
	ToString	Returns a reference string that represents the range.

[Top](#)

See Also

Reference

[XLCeRange Class](#)

[C1.C1Excel Namespace](#)

XLCeRange Constructor

Overload List

Overload	Description
XLCeRange Constructor()	Creates an instance of an

	XLCellRange object containing an empty range.
XLCellRange Constructor(Int32,Int32,Int32,Int32)	Creates an instance of an XLCellRange object containing a specified range.
XLCellRange Constructor(XLSheet,Int32,Int32,Int32,Int32)	Creates an instance of an XLCellRange object containing a specified range.

<p>XLCellRange</p> <p>Constructor(XLSheet,Int32,Int32,Int32,Int32,XLReferenceType,XLReferenceType,XLReferenceType,XLReferenceType)</p>	<p>Creates an instance of an XLCellRange object containing a specified range.</p>
<p>XLCellRange Constructor(XLSheet,Int32,Int32,XLReferenceType,XLReferenceType)</p>	<p>Creates an instance of an XLCellRange object containing a specified range.</p>
<p>XLCellRange</p> <p>Constructor(Int32,Int32,Int32,Int32,Int32,Int32,XLReferenceType,XLReferenceType,XLReferenceType,XLReferenceType)</p>	<p>Creates an instance of an XLCellRange object containing</p>

	ng a specifie d range.
XLCellRange Constructor(XLSheet,String)	Creates an instance of an XLCellR ange object containi ng a specifie d range.
XLCellRange Constructor(C1XLBook,String)	Creates an instance of an XLCellR ange object containi ng a specifie d range.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCeIlRange Class](#)

[XLCeIlRange Members](#)

XLCeIlRange Constructor()

Creates an instance of an [XLCeIlRange](#) object containing an empty range.

Syntax

Visual Basic (Declaration)	
Public Function New()	
C#	
public XLCeIlRange()	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCeIlRange Class](#)

[XLCeIlRange Members](#)

[Overload List](#)

XLCeIlRange Constructor(Int32,Int32,Int32,Int32)

Top row in the range.

Bottom row in the range.

Left column in the range.

Right column in the range.

Creates an instance of an [XLCeIlRange](#) object containing a specified range.

Syntax

Visual Basic (Declaration)

```
Public Function New( _  
    ByVal rowFrom As System.Integer, _  
    ByVal rowTo As System.Integer, _  
    ByVal colFrom As System.Integer, _  
    ByVal colTo As System.Integer _  
)
```

C#

```
public XLCellRange(  
    System.int rowFrom,  
    System.int rowTo,  
    System.int colFrom,  
    System.int colTo  
)
```

Parameters

rowFrom

Top row in the range.

rowTo

Bottom row in the range.

colFrom

Left column in the range.

colTo

Right column in the range.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCeLLRange Class](#)
[XLCeLLRange Members](#)
[Overload List](#)

XLCeLLRange Constructor(XLSheet,Int32,Int32,Int32,Int32)

The [XLSheet](#) of this cell range.

Top row in the range.

Bottom row in the range.

Left column in the range.

Right column in the range.

Creates an instance of an [XLCeLLRange](#) object containing a specified range.

Syntax

Visual Basic (Declaration)

```
Public Function New( _  
    ByVal sheet As XLSheet, _  
    ByVal rowFrom As System.Integer, _  
    ByVal rowTo As System.Integer, _  
    ByVal colFrom As System.Integer, _  
    ByVal colTo As System.Integer _  
)
```

C#

```
public XLCeLLRange(  
    XLSheet sheet,  
    System.int rowFrom,  
    System.int rowTo,  
    System.int colFrom,  
    System.int colTo  
)
```

Parameters

sheet

The [XLSheet](#) of this cell range.

rowFrom

Top row in the range.

rowTo

Bottom row in the range.

colFrom

Left column in the range.

colTo

Right column in the range.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)
[XLCellRange Members](#)
[Overload List](#)

XLCellRange

Constructor(XLSheet,Int32,Int32,Int32,Int32,XLReferenceType,XLReferenceType,XLReferenceType,XLReferenceType)

The [XLSheet](#) of this cell range.

Top row in the range.

Bottom row in the range.

Left column in the range.

Right column in the range.

The reference type of the top row in the range.

The reference type of the bottom row in the range.

The reference type of the left column in the range.

The reference type of the right column in the range.

Creates an instance of an [XLCellRange](#) object containing a specified range.

Syntax

Visual Basic (Declaration)

```
Public Function New( _  
    ByVal sheet As XLSheet, _  
    ByVal rowFrom As System.Integer, _  
    ByVal rowTo As System.Integer, _  
    ByVal colFrom As System.Integer, _  
    ByVal colTo As System.Integer, _  
    ByVal rowFromRef As XLReferenceType, _  
    ByVal rowToRef As XLReferenceType, _  
    ByVal colFromRef As XLReferenceType, _  
    ByVal colToRef As XLReferenceType _  
)
```

C#

```
public XLCellRange(  
    XLSheet sheet,  
    System.int rowFrom,  
    System.int rowTo,  
    System.int colFrom,  
    System.int colTo,  
    XLReferenceType rowFromRef,  
    XLReferenceType rowToRef,  
    XLReferenceType colFromRef,  
    XLReferenceType colToRef  
)
```

Parameters

sheet

The [XLSheet](#) of this cell range.

rowFrom

Top row in the range.

rowTo

Bottom row in the range.

colFrom

Left column in the range.

colTo

Right column in the range.

rowFromRef

The reference type of the top row in the range.

rowToRef

The reference type of the bottom row in the range.

colFromRef

The reference type of the left column in the range.

colToRef

The reference type of the right column in the range.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)
[XLCellRange Members](#)
[Overload List](#)

XLCellRange Constructor(XLSheet,Int32,Int32,XLReferenceType,XLReferenceType)
The [XLSheet](#) of the range.

The row in the range equal one cell.

The column in the range equal one cell.

The reference type of the row.

The reference type of the column.

Creates an instance of an [XLCellRange](#) object containing a specified range.

Syntax

Visual Basic (Declaration)

```
Public Function New( _  
    ByVal ws As XLSheet, _  
    ByVal row As System.Integer, _  
    ByVal col As System.Integer, _  
    ByVal rowRef As XLReferenceType, _  
    ByVal colRef As XLReferenceType _  
)
```

C#

```
public XLCellRange(  
    XLSheet ws,  
    System.int row,  
    System.int col,  
    XLReferenceType rowRef,  
    XLReferenceType colRef  
)
```

Parameters

ws

The [XLSheet](#) of the range.

row

The row in the range equal one cell.

col

The column in the range equal one cell.

rowRef

The reference type of the row.

colRef

The reference type of the column.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)
[XLCellRange Members](#)
[Overload List](#)

XLCellRange

Constructor(Int32,Int32,Int32,Int32,Int32,Int32,XLReferenceType,XLReferenceType,XLReferenceType,XLReferenceType)

First index of the [XLSheet](#) of the range.

Last index of the [XLSheet](#) of the range.

Top row in the range.

Bottom row in the range.

Left column in the range.

Right column in the range.

The reference type of the top row in the range.

The reference type of the bottom row in the range.

The reference type of the left column in the range.

The reference type of the right column in the range.

Creates an instance of an [XLCellRange](#) object containing a specified range.

Syntax

Visual Basic (Declaration)

```
Public Function New( _  
    ByVal sheetFrom As System.Integer, _  
    ByVal sheetTo As System.Integer, _  
    ByVal rowFrom As System.Integer, _  
    ByVal rowTo As System.Integer, _  
    ByVal colFrom As System.Integer, _  
    ByVal colTo As System.Integer, _  
    ByVal rowFromRef As XLReferenceType, _  
    ByVal rowToRef As XLReferenceType, _  
    ByVal colFromRef As XLReferenceType, _  
    ByVal colToRef As XLReferenceType _  
)
```

C#

```
public XLCellRange(  
    System.int sheetFrom,  
    System.int sheetTo,  
    System.int rowFrom,  
    System.int rowTo,  
    System.int colFrom,  
    System.int colTo,  
    XLReferenceType rowFromRef,  
    XLReferenceType rowToRef,  
    XLReferenceType colFromRef,  
    XLReferenceType colToRef  
)
```

Parameters

sheetFrom

First index of the [XLSheet](#) of the range.

sheetTo

Last index of the [XLSheet](#) of the range.

rowFrom

Top row in the range.

rowTo

Bottom row in the range.

colFrom

Left column in the range.

colTo

Right column in the range.

rowFromRef

The reference type of the top row in the range.

rowToRef

The reference type of the bottom row in the range.

colFromRef

The reference type of the left column in the range.

colToRef

The reference type of the right column in the range.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)
[XLCellRange Members](#)
[Overload List](#)

XLCellRange Constructor(XLSheet,String)

The owner sheet for the range.

The text presentation of a specified range without sheets.

Creates an instance of an [XLCellRange](#) object containing a specified range.

Syntax

Visual Basic (Declaration)	
<pre>Public Function New(_ ByVal sheet As XLSheet, _ ByVal reference As System.String _)</pre>	
C#	
<pre>public XLCellRange(XLSheet sheet, System.string reference)</pre>	

Parameters

sheet

The owner sheet for the range.

reference

The text presentation of a specified range without sheets.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)
[XLCellRange Members](#)
[Overload List](#)

XLCellRange Constructor(C1XLBook,String)

The owner workbook for the range.

The text presentation of a specified range.

Creates an instance of an [XLCellRange](#) object containing a specified range.

Syntax

Visual Basic (Declaration)

```
Public Function New( _  
    ByVal book As C1XLBook, _  
    ByVal reference As System.String _  
)
```

C#

```
public XLCellRange(  
    C1XLBook book,  
    System.string reference  
)
```

Parameters

book

The owner workbook for the range.

reference

The text presentation of a specified range.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)
[XLCellRange Members](#)
[Overload List](#)

Methods

For a list of all members of this type, see [XLCellRange members](#).

Public Methods

	Name	Description
≡	Clone	Creates a copy of this cell range.
≡	Contains	Overloaded. Determines whether the range contains a specific cell.
≡	Intersects	Determines whether the range intersects another range.
≡	ToString	Returns a reference string that represents the range.

[Top](#)

See Also

Reference

[XLCellRange Class](#)
[C1.C1Excel Namespace](#)

Clone Method

Creates a copy of this cell range.

Syntax

Visual Basic (Declaration)	
Public Function Clone() As XLCellRange	
C#	
public XLCellRange Clone()	

Return Value

A new [XLCellRange](#) object that is a copy of the current instance.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

Contains Method

Determines whether the range contains a specific cell.

Overload List

Overload	Description
Contains(XLSheet,Int32,Int32)	Determines whether the range contains a specific cell.
Contains(XLCellRange)	Determines whether the range contains a specific range.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

Contains(XLSheet,Int32,Int32) Method

Determines whether the range contains a specific cell.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Contains( _  
    ByVal sheet As XLSheet, _  
    ByVal row As System.Integer, _  
    ByVal col As System.Integer _  
) As System.Boolean
```

C#

```
public System.bool Contains(  
    XLSheet sheet,  
    System.int row,  
    System.int col  
)
```

Parameters

sheet

row

col

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

[Overload List](#)

Contains(XLCellRange) Method

Determines whether the range contains a specific range.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Contains( _  
    ByVal cr As XLCellRange _  
) As System.Boolean
```

C#

```
public System.bool Contains(  
    XLCellRange cr  
)
```

Parameters

cr

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)
[XLCellRange Members](#)
[Overload List](#)

Intersects Method

Determines whether the range intersects another range.

Syntax

Visual Basic (Declaration)

```
Public Function Intersects( _  
    ByVal cr As XLCellRange _  
) As System.Boolean
```

C#

```
public System.bool Intersects(  

```

```
    XLCeIlRange cr  
)
```

Parameters

cr

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCeIlRange Class](#)
[XLCeIlRange Members](#)

ToString Method

Returns a reference string that represents the range.

Syntax

Visual Basic (Declaration)	
<pre>Public Overrides Function ToString() As System.String</pre>	
C#	
<pre>public override System.string ToString()</pre>	

Return Value

A reference string that represents the range.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference














[XLCellRange Class](#)







[XLCellRange Members](#)

Properties

For a list of all members of this type, see [XLCellRange members](#).

Public Properties

	Name	Description
	Book	Gets a reference to the parent C1XLBook object.
	ColumnCount	Gets the number of columns in the range.
	ColumnFrom	Gets the index of the left column in the range.
	ColumnFromRef	Gets the reference type of the left column in the range.
	ColumnTo	Gets or sets the index of the last column in the range.
	ColumnToRef	Gets the reference type of the right column in the range.
	IsEmpty	Determines whether the range is empty.
	RangeType	Gets or sets a type (default, indirect or offset) for this cell range.
	RowCount	Gets the number of rows in the range.
	RowFrom	Gets the index of the top row in the range.
	RowFromRef	Gets the reference type of the top row in the range.
	RowTo	Gets or sets the index of the last row in the range.
	RowToRef	Gets the reference type of the bottom row in the range.

	SheetCount	Gets the number of sheets in the range.
	SheetFrom	Gets the index of the first sheet in the range.
	Sheets	Gets an array of XLSheet objects.
	SheetTo	Gets or sets the index of the last sheet in the range.
	Style	Gets or sets the XLStyle object associated with this range of cells.
	Value	Gets or sets the complex value associated with this range of cells.

[Top](#)

See Also

Reference

[XLCellRange Class](#)

[C1.C1Excel Namespace](#)

Book Property

Gets a reference to the parent [C1XLBook](#) object.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Book As C1XLBook</code>	
C#	
<code>public C1XLBook Book {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

ColumnCount Property

Gets the number of columns in the range.

Syntax

Visual Basic (Declaration)	
Public ReadOnly Property ColumnCount As System.Integer	
C#	
public System.int ColumnCount { get ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

ColumnFrom Property

Gets the index of the left column in the range.

Syntax

Visual Basic (Declaration)	
Public ReadOnly Property ColumnFrom As System.Integer	
C#	
public System.int ColumnFrom { get ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

ColumnFromRef Property

Gets the reference type of the left column in the range.

Syntax

Visual Basic (Declaration)	
Public ReadOnly Property ColumnFromRef As XLReferenceType	
C#	
public XLReferenceType ColumnFromRef {get;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

ColumnTo Property

Gets or sets the index of the last column in the range.

Syntax

Visual Basic (Declaration)	
----------------------------	--

Public Property ColumnTo As System.Integer
C#
public System.int ColumnTo {get; set;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

ColumnToRef Property

Gets the reference type of the right column in the range.

Syntax

Visual Basic (Declaration)
Public ReadOnly Property ColumnToRef As XLReferenceType
C#
public XLReferenceType ColumnToRef {get;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

IsEmpty Property

Determines whether the range is empty.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property IsEmpty As System.Boolean
```

C#

```
public System.Boolean IsEmpty {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

RangeType Property

Gets or sets a type (default, indirect or offset) for this cell range.

Syntax

Visual Basic (Declaration)

```
Public Property RangeType As XLRangeType
```

C#

```
public XLRangeType RangeType {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

RowCount Property

Gets the number of rows in the range.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property RowCount As System.Integer</code>	
C#	
<code>public System.int RowCount {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

RowFrom Property

Gets the index of the top row in the range.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property RowFrom As System.Integer</code>	
C#	

```
public System.int RowFrom {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

RowFromRef Property

Gets the reference type of the top row in the range.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property RowFromRef As XLReferenceType
```

C#

```
public XLReferenceType RowFromRef {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

RowTo Property

Gets or sets the index of the last row in the range.

Syntax

Visual Basic (Declaration)	
<code>Public Property RowTo As System.Integer</code>	
C#	
<code>public System.int RowTo {get; set;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

RowToRef Property

Gets the reference type of the bottom row in the range.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property RowToRef As XLReferenceType</code>	
C#	
<code>public XLReferenceType RowToRef {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

SheetCount Property

Gets the number of sheets in the range.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property SheetCount As System.Integer
```

C#

```
public System.int SheetCount {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

SheetFrom Property

Gets the index of the first sheet in the range.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property SheetFrom As System.Integer
```

C#

```
public System.int SheetFrom {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

Sheets Property

Gets an array of [XLSheet](#) objects.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Sheets As XLSheet()</code>	
C#	
<code>public XLSheet[] Sheets {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

SheetTo Property

Gets or sets the index of the last sheet in the range.

Syntax

Visual Basic (Declaration)	
----------------------------	--

Public Property SheetTo As System.Integer

C#

public System.int SheetTo {get; set;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

Style Property

Gets or sets the [XLStyle](#) object associated with this range of cells.

Syntax

Visual Basic (Declaration)

Public Property Style As XLStyle

C#

public XLStyle Style {get; set;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

Value Property

Gets or sets the complex value associated with this range of cells.

Syntax

Visual Basic (Declaration)

```
Public Property Value As System.Object
```

C#

```
public System.Object Value {get; set;}
```

Remarks

The value may contain strings, numeric, Boolean, DateTime, or null objects. Other types of objects cannot be saved in Excel files.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRange Class](#)

[XLCellRange Members](#)

XLCellRangeCollection

Represents a collection of [XLCellRange](#) objects on an [XLSheet](#).

Object Model

XLCellRangeCollection

Syntax

Visual Basic (Declaration)

```
<System.Reflection.DefaultMemberAttribute("Item")>
```

Public Class XLCellRangeCollection

C#

[System.Reflection.DefaultMemberAttribute("Item")]
--

public class XLCellRangeCollection

Inheritance Hierarchy

System.Object

C1.C1Excel.XLCellRangeCollection

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRangeCollection Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents a collection of [XLCellRange](#) objects on an [XLSheet](#).

Object Model

XLCellRangeCollection

Syntax

Visual Basic (Declaration)

<System.Reflection.DefaultMemberAttribute("Item")>
--

Public Class XLCellRangeCollection

C#

[System.Reflection.DefaultMemberAttribute("Item")]
--

public class XLCellRangeCollection

Inheritance Hierarchy

System.Object

C1.C1Excel.XLCellRangeCollection

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRangeCollection Members](#)






[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)

The following tables list the members exposed by [XLCellRangeCollection](#).

Public Properties

	Name	Description
	ActiveIndex	Gets or sets the active index in the collection.
	Book	Gets a reference to the parent C1XLBook object.
	Count	Gets the number of XLCellRange objects in the collection.
	Item	Gets the XLCellRange object at the specified position in the collection.
	Sheet	Gets the XLSheet object that owns the collection.

[Top](#)

Public Methods

	Name	Description
≡	Add	Overloaded. Appends an XLCeIlRange object to the collection.
≡	Clear	Removes all XLCeIlRange objects from the collection.
≡	Contains	Checks whether the collection contains a specific XLCeIlRange object.
≡	IndexOf	Gets the position of an XLCeIlRange object in the collection.
≡	Insert	Inserts an XLCeIlRange object at a specific position in the collection.
≡	Remove	Removes an XLCeIlRange object from the collection.
≡	RemoveAt	Removes an XLCeIlRange object at a specific position from the collection.

[Top](#)

See Also

Reference

[XLCeIlRangeCollection Class](#)





[C1.C1Excel Namespace](#)

Methods

For a list of all members of this type, see [XLCeIlRangeCollection members](#).

Public Methods

	Name	Description
≡	Add	Overloaded. Appends an XLCeIlRange object to the collection.
≡	Clear	Removes all XLCeIlRange objects from the collection.
≡	Contains	Checks whether the collection contains a specific XLCeIlRange object.

 IndexOf	Gets the position of an XLCellRange object in the collection.
 Insert	Inserts an XLCellRange object at a specific position in the collection.
 Remove	Removes an XLCellRange object from the collection.
 RemoveAt	Removes an XLCellRange object at a specific position from the collection.

[Top](#)

See Also

Reference

[XLCellRangeCollection Class](#)

[C1.C1Excel Namespace](#)

Add Method

Appends an [XLCellRange](#) object to the collection.

Overload List

Overload	Description
Add(XLCellRange)	Appends an XLCellRange object to the collection.
Add(Int32,Int32,Int32,Int32)	Creates an XLCellRange object and appends it to the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRangeCollection Class](#)

[XLCellRangeCollection Members](#)

Add(XLCellRange) Method

[XLCellRange](#) object to add to the collection.

Appends an [XLCellRange](#) object to the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function Add(_ ByVal cr As XLCellRange _) As XLCellRange</pre>	
C#	
<pre>public XLCellRange Add(XLCellRange cr)</pre>	

Parameters

cr

[XLCellRange](#) object to add to the collection.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added to the collection (usually because it overlaps another cell range already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRangeCollection Class](#)

[XLCellRangeCollection Members](#)

[Overload List](#)

Add(Int32,Int32,Int32,Int32) Method

Index of the top row in the cell range.

Index of the left column in the cell range.

Number of rows in the cell range.

Number of columns in the cell range.

Creates an [XLCellRange](#) object and appends it to the collection.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Add( _  
    ByVal rowIndex As System.Integer, _  
    ByVal colIndex As System.Integer, _  
    ByVal rowCount As System.Integer, _  
    ByVal colCount As System.Integer _  
) As XLCellRange
```

C#

```
public XLCellRange Add(  
    System.int rowIndex,  
    System.int colIndex,  
    System.int rowCount,  
    System.int colCount  
)
```

Parameters

rowIndex

Index of the top row in the cell range.

colIndex

Index of the left column in the cell range.

rowCount

Number of rows in the cell range.

colCount

Number of columns in the cell range.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added to the collection (usually because it overlaps another cell range already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRangeCollection Class](#)
[XLCellRangeCollection Members](#)
[Overload List](#)

Clear Method

Removes all [XLCellRange](#) objects from the collection.

Syntax

Visual Basic (Declaration)	
Public Sub Clear()	
C#	
public void Clear()	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCeLLRangeCollection Class](#)

[XLCeLLRangeCollection Members](#)

Contains Method

[XLCeLLRange](#) object to look for.

Checks whether the collection contains a specific [XLCeLLRange](#) object.

Syntax

Visual Basic (Declaration)	
<pre>Public Function Contains(_ ByVal cr As XLCeLLRange _) As System.Boolean</pre>	
C#	
<pre>public System.bool Contains(XLCeLLRange cr)</pre>	

Parameters

cr

[XLCeLLRange](#) object to look for.

Return Value

True if the collection contains the range, False otherwise.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCeLLRangeCollection Class](#)

[XLCeLLRangeCollection Members](#)

IndexOf Method

[XLCellRange](#) object to look for.

Gets the position of an [XLCellRange](#) object in the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Function IndexOf(_ ByVal cr As XLCellRange _) As System.Integer</pre>	
C#	
<pre>public System.int IndexOf(XLCellRange cr)</pre>	

Parameters

cr

[XLCellRange](#) object to look for.

Return Value

The position of the object in the collection, or -1 if the object is not a member of the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRangeCollection Class](#)

[XLCellRangeCollection Members](#)

Insert Method

Position where the object will be inserted.

Object to insert in the collection.

Inserts an [XLCellRange](#) object at a specific position in the collection.

Syntax

Visual Basic (Declaration)

```
Public Function Insert( _  
    ByVal index As System.Integer, _  
    ByVal cr As XLCellRange _  
) As XLCellRange
```

C#

```
public XLCellRange Insert(  
    System.int index,  
    XLCellRange cr  
)
```

Parameters

index

Position where the object will be inserted.

cr

Object to insert in the collection.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added to the collection (usually because it overlaps another cell range already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

Remove Method

[XLCellRange](#) object to remove from the collection.

Removes an [XLCellRange](#) object from the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Sub Remove(_ ByVal cr As XLCellRange _)</pre>	
C#	
<pre>public void Remove(XLCellRange cr)</pre>	

Parameters

cr

[XLCellRange](#) object to remove from the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

RemoveAt Method

Index of the object to remove from the collection.

Removes an [XLCellRange](#) object at a specific position from the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Sub RemoveAt(_ ByVal index As System.Integer _)</pre>	
C#	
<pre>public void RemoveAt(System.int index)</pre>	

Parameters

index

Index of the object to remove from the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference


[XLCellRangeCollection Class](#)





[XLCellRangeCollection Members](#)

Properties

For a list of all members of this type, see [XLCellRangeCollection members](#).

Public Properties

	Name	Description
	ActiveIndex	Gets or sets the active index in the collection.

	Book	Gets a reference to the parent C1XLBook object.
	Count	Gets the number of XLCellRange objects in the collection.
	Item	Gets the XLCellRange object at the specified position in the collection.
	Sheet	Gets the XLSheet object that owns the collection.

[Top](#)

See Also

Reference

[XLCellRangeCollection Class](#)

[C1.C1Excel Namespace](#)

ActiveIndex Property

Gets or sets the active index in the collection.

Syntax

Visual Basic (Declaration)	
Public Property ActiveIndex As System.Integer	
C#	
public System.int ActiveIndex { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRangeCollection Class](#)

[XLCellRangeCollection Members](#)

Book Property

Gets a reference to the parent [C1XLBook](#) object.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property Book As C1XLBook
```

C#

```
public C1XLBook Book {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRangeCollection Class](#)

[XLCellRangeCollection Members](#)

Count Property

Gets the number of [XLCellRange](#) objects in the collection.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property Count As System.Integer
```

C#

```
public System.int Count {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRangeCollection Class](#)

[XLCellRangeCollection Members](#)

Item Property

Gets the [XLCellRange](#) object at the specified position in the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public ReadOnly Default Property Item(_ ByVal index As System.Integer _) As XLCellRange</pre>	
C#	
<pre>public XLCellRange this[System.int index]; {get;}</pre>	

Parameters

index

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRangeCollection Class](#)

[XLCellRangeCollection Members](#)

Sheet Property

Gets the [XLSheet](#) object that owns the collection.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Sheet As XLSheet</code>	
C#	
<code>public XLSheet Sheet {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCellRangeCollection Class](#)

[XLCellRangeCollection Members](#)

XLColumn

Represents a column in a worksheet. This class provides properties for setting the column's width, style, and visibility.

Object Model

XLColumn

Syntax

Visual Basic (Declaration)	
<code>Public Class XLColumn</code>	
C#	
<code>public class XLColumn</code>	

Remarks

The [XLColumn](#) objects do not contain any data. If you remove a column from the collection, the data will be lost. If you later re-insert that same column back into the collection, the column will be blank.

Inheritance Hierarchy

System.Object

C1.C1Excel.XLColumn

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumn Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents a column in a worksheet. This class provides properties for setting the column's width, style, and visibility.

Object Model

XLColumn

Syntax

Visual Basic (Declaration)	
Public Class XLColumn	
C#	
public class XLColumn	

Remarks

The [XLColumn](#) objects do not contain any data. If you remove a column from the collection, the data will be lost. If you later re-insert that same column back into the collection, the column will be blank.

Inheritance Hierarchy

System.Object

C1.C1Excel.XLColumn

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumn Members](#)


[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)





The following tables list the members exposed by [XLColumn](#).






Public Constructors

	Name	Description
	XLColumn Constructor	Creates a new instance of the XLColumn class.

[Top](#)


Public Properties

	Name	Description
	Book	Gets a reference to the parent C1XLBook object.
	Collapsed	Gets or sets collapsed flag for the column.
	IsSubtotal	Gets whether the column is subtotal.
	OutlineLevel	Gets or sets subtotal outline level for the column.

	PageBreak	Gets or sets whether there will be a forced page break after this column.
	Sheet	Gets a reference to the parent XLSheet object.
	Style	Gets or sets the XLStyle object that determines the appearance of the column.
	Visible	Gets or sets whether the column is visible.
	Width	Gets or sets the width of the column, in twips.

[Top](#)

Public Methods

	Name	Description
	Clone	Creates a new XLColumn object that is a copy of the current instance.

[Top](#)

See Also

Reference

[XLColumn Class](#)

[C1.C1Excel Namespace](#)

XLColumn Constructor

Creates a new instance of the [XLColumn](#) class.

Syntax

Visual Basic (Declaration)	
Public Function New()	
C#	
public XLColumn()	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference


[XLColumn Class](#)

[XLColumn Members](#)

Methods

For a list of all members of this type, see [XLColumn members](#).

Public Methods

	Name	Description
	Clone	Creates a new XLColumn object that is a copy of the current instance.

[Top](#)

See Also

Reference

[XLColumn Class](#)

[C1.C1Excel Namespace](#)

Clone Method

Creates a new [XLColumn](#) object that is a copy of the current instance.

Syntax

Visual Basic (Declaration)	
Public Function Clone() As XLColumn	
C#	
public XLColumn Clone()	

Return Value

A new [XLColumn](#) object that is a copy of the current instance.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference









[XLColumn Class](#)


[XLColumn Members](#)

Properties

For a list of all members of this type, see [XLColumn members](#).

Public Properties

	Name	Description
	Book	Gets a reference to the parent C1XLBook object.
	Collapsed	Gets or sets collapsed flag for the column.
	IsSubtotal	Gets whether the column is subtotal.
	OutlineLevel	Gets or sets subtotal outline level for the column.
	PageBreak	Gets or sets whether there will be a forced page break after this column.
	Sheet	Gets a reference to the parent XLSheet object.
	Style	Gets or sets the XLStyle object that determines the appearance of the column.
	Visible	Gets or sets whether the column is visible.

	Width	Gets or sets the width of the column, in twips.
---	--------------	---

[Top](#)

See Also

Reference

[XLColumn Class](#)

[C1.C1Excel Namespace](#)

Book Property

Gets a reference to the parent [C1XLBook](#) object.

Syntax

Visual Basic (Declaration)	
Public ReadOnly Property Book As C1XLBook	
C#	
public C1XLBook Book { get ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumn Class](#)

[XLColumn Members](#)

Collapsed Property

Gets or sets collapsed flag for the column.

Syntax

Visual Basic (Declaration)	
----------------------------	--

Public Property Collapsed As System.Boolean

C#

public System.bool Collapsed {get; set;}
--

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumn Class](#)

[XLColumn Members](#)

IsSubtotal Property

Gets whether the column is subtotal.

Syntax

Visual Basic (Declaration)

Public ReadOnly Property IsSubtotal As System.Boolean

C#

public System.bool IsSubtotal {get;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumn Class](#)

[XLColumn Members](#)

OutlineLevel Property

Gets or sets subtotal outline level for the column.

Syntax

Visual Basic (Declaration)	
Public Property OutlineLevel As System.Integer	
C#	
public System.int OutlineLevel { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumn Class](#)

[XLColumn Members](#)

PageBreak Property

Gets or sets whether there will be a forced page break after this column.

Syntax

Visual Basic (Declaration)	
Public Property PageBreak As System.Boolean	
C#	
public System.bool PageBreak { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumn Class](#)

[XLColumn Members](#)

Sheet Property

Gets a reference to the parent [XLSheet](#) object.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Sheet As XLSheet</code>	
C#	
<code>public XLSheet Sheet {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumn Class](#)

[XLColumn Members](#)

Style Property

Gets or sets the [XLStyle](#) object that determines the appearance of the column.

Syntax

Visual Basic (Declaration)	
<code>Public Property Style As XLStyle</code>	
C#	

```
public XLStyle Style {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumn Class](#)

[XLColumn Members](#)

Visible Property

Gets or sets whether the column is visible.

Syntax

Visual Basic (Declaration)	
<code>Public Property Visible As System.Boolean</code>	
C#	
<code>public System.bool Visible {get; set;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumn Class](#)

[XLColumn Members](#)

Width Property

Gets or sets the width of the column, in twips.

Syntax

Visual Basic (Declaration)	
<code>Public Property Width As System.Integer</code>	
C#	
<code>public System.int Width {get; set;}</code>	

Remarks

A value of -1 indicates that the column should be displayed using the sheet's [XLSheet.DefaultColumnWidth](#).

To convert between pixels and twips, use the [C1XLBook.TwipsToPixels](#) and [PixelsToTwips](#) methods.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumn Class](#)

[XLColumn Members](#)

XLColumnCollection

[Example](#)

Represents a collection of [XLColumn](#) objects that represent the individual columns in each [XLSheet](#).

Object Model

XLColumnCollection

Syntax

Visual Basic (Declaration)	
<code><System.Reflection.DefaultMemberAttribute("Item")></code>	
<code>Public Class XLColumnCollection</code>	

C#

```
[System.Reflection.DefaultMemberAttribute("Item")]  
public class XLColumnCollection
```

Remarks

The collection has methods for counting, enumerating, adding, and removing columns from the collection.

The [XLColumn](#) objects do not contain any data. If you remove a column from the collection, the data will be lost. If you later re-insert that same column back into the collection, the column will be blank.

Example

Note that you can create columns automatically by using the sheet's indexer. For example, the following code retrieves the cell at coordinates (3,3) and in doing so automatically creates four rows and four columns automatically:

- [C#](#)

```
C1XLBook book = new C1XLBook();  
XLSheet sheet = book.Sheets[0];  
XLCell cell = sheet[3,3]; // creates 4 rows and 4 columns
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLColumnCollection

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Members](#)

[C1.C1Excel Namespace](#)

Overview

[Example](#)

Represents a collection of [XLColumn](#) objects that represent the individual columns in each [XLSheet](#).

Object Model

XLColumnCollection

Syntax

Visual Basic (Declaration)	
<pre><System.Reflection.DefaultMemberAttribute("Item")> Public Class XLColumnCollection</pre>	
C#	
<pre>[System.Reflection.DefaultMemberAttribute("Item")] public class XLColumnCollection</pre>	

Remarks

The collection has methods for counting, enumerating, adding, and removing columns from the collection.

The [XLColumn](#) objects do not contain any data. If you remove a column from the collection, the data will be lost. If you later re-insert that same column back into the collection, the column will be blank.

Example

Note that you can create columns automatically by using the sheet's indexer. For example, the following code retrieves the cell at coordinates (3,3) and in doing so automatically creates four rows and four columns automatically:

- [C#](#)

```
C1XLBook book = new C1XLBook();
XLSheet sheet = book.Sheets[0];
XLCell cell = sheet[3,3]; // creates 4 rows and 4 columns
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLColumnCollection

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Members](#)






[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)


The following tables list the members exposed by [XLColumnCollection](#).







Public Properties

	Name	Description
	Book	Gets a reference to the parent C1XLBook object.
	Count	Gets the number of items in the collection.
	Frozen	Gets or sets the number of frozen columns in the collection.
	Item	Returns a reference to the XLColumn object at the specified index.
	Sheet	Gets a reference to the parent XLSheet object.

[Top](#)

Public Methods

	Name	Description
	Add	Overloaded. Creates a new XLColumn object and adds it to the collection.

≡  Clear	Removes all items from the collection.
≡  Contains	Determines whether an XLColumn is a member of the collection.
≡  IndexOf	Gets the index of a given XLRow object in the collection.
≡  Insert	Overloaded. Creates a new XLColumn object and inserts it at a specific position in the collection.
≡  Remove	Removes an XLColumn object from the collection.
≡  RemoveAt	Removes the XLColumn object at a given position from the collection.

[Top](#)

See Also








Reference

[XLColumnCollection Class](#)

[C1.C1Excel Namespace](#)

Methods

>

Name	Description
≡  Add	Overloaded. Creates a new XLColumn object and adds it to the collection.
≡  Clear	Removes all items from the collection.
≡  Contains	Determines whether an XLColumn is a member of the collection.
≡  IndexOf	Gets the index of a given XLRow object in the collection.
≡  Insert	Overloaded. Creates a new XLColumn object and inserts it at a specific position in the collection.
≡  Remove	Removes an XLColumn object from the collection.
≡  RemoveAt	Removes the XLColumn object at a given position from the collection.

[Top](#)

See Also

Reference

[XLColumnCollection Class](#)

[C1.C1Excel Namespace](#)

Add Method

Creates a new [XLColumn](#) object and adds it to the collection.

Overload List

Overload	Description
Add()	Creates a new XLColumn object and adds it to the collection.
Add(XLColumn)	Adds an XLColumn object to the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)

[XLColumnCollection Members](#)

Add() Method

Creates a new [XLColumn](#) object and adds it to the collection.

Syntax

Visual Basic (Declaration)	
Public Overloads Function Add() As XLColumn	

C#

```
public XLColumn Add()
```

Return Value

A reference to the new [XLColumn](#) object.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)

[XLColumnCollection Members](#)

[Overload List](#)

Add(XLColumn) Method

The item to add to the collection.

Adds an [XLColumn](#) object to the collection.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Add( _  
    ByVal col As XLColumn _  
) As XLColumn
```

C#

```
public XLColumn Add(  
    XLColumn col  
)
```

Parameters

col

The item to add to the collection.

Return Value

A reference to the item that was added to the collection (in this case, always the [col](#) parameter).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)

[XLColumnCollection Members](#)

[Overload List](#)

Clear Method

Removes all items from the collection.

Syntax

Visual Basic (Declaration)	
Public Sub Clear()	
C#	
public void Clear()	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)

[XLColumnCollection Members](#)

Contains Method

Item to look for.

Determines whether an [XLColumn](#) is a member of the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Function Contains(_ ByVal col As XLColumn _) As System.Boolean</pre>	
C#	
<pre>public System.bool Contains(XLColumn col)</pre>	

Parameters

col

Item to look for.

Return Value

True if the collection contains the item, False otherwise.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)

[XLColumnCollection Members](#)

IndexOf Method

Item to look for.

Gets the index of a given [XLRow](#) object in the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Function IndexOf(_ ByVal col As XLColumn _) As System.Integer</pre>	
C#	
<pre>public System.int IndexOf(XLColumn col)</pre>	

Parameters

col

Item to look for.

Return Value

The position of the item in the collection, or -1 if the item is not a member of the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)

[XLColumnCollection Members](#)

Insert Method

Creates a new [XLColumn](#) object and inserts it at a specific position in the collection.

Overload List

Overload	Description
Insert(Int32)	Creates a new XLColumn object and inserts it at a specific position in the collection.
Insert(Int32,XLColumn)	Inserts an XLColumn object at a specific position in the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)

[XLColumnCollection Members](#)

Insert(Int32) Method

Position where the new item will be inserted.

Creates a new [XLColumn](#) object and inserts it at a specific position in the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function Insert(_ ByVal index As System.Integer _) As XLColumn</pre>	
C#	
<pre>public XLColumn Insert(System.int index)</pre>	

Parameters

index

Position where the new item will be inserted.

Return Value

A reference to the new item.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)

[XLColumnCollection Members](#)

[Overload List](#)

Insert(Int32,XLColumn) Method

Position where the item will be inserted.

Item that will be inserted.

Inserts an [XLColumn](#) object at a specific position in the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function Insert(_ ByVal index As System.Integer, _ ByVal col As XLColumn _) As XLColumn</pre>	
C#	
<pre>public XLColumn Insert(System.int index, XLColumn col)</pre>	

Parameters

index

Position where the item will be inserted.

col

Item that will be inserted.

Return Value

A reference to the item that was added to the collection.

Remarks

The maximum number of [XLColumn](#) objects in an [XLSheet](#) is 256. This is a limitation imposed by Excel 2003 and below.

For Excel 2007 and above, the maximum number of [XLColumn](#) objects in an [XLSheet](#) is 18,278.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)

[XLColumnCollection Members](#)

[Overload List](#)

Remove Method

Item to be removed from the collection.

Removes an [XLColumn](#) object from the collection.

Syntax

Visual Basic (Declaration)	
----------------------------	--

```
Public Function Remove( _
    ByVal col As XLColumn _
) As XLColumn
```

C#

```
public XLColumn Remove(
    XLColumn col
)
```

Parameters

col

Item to be removed from the collection.

Return Value

A reference to the item that was removed.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)

[XLColumnCollection Members](#)

RemoveAt Method

Index of the item to remove from the collection.

Removes the [XLColumn](#) object at a given position from the collection.

Syntax

Visual Basic (Declaration)

```
Public Function RemoveAt( _
    ByVal index As System.Integer _
) As XLColumn
```

C#

```
public XLColumn RemoveAt(  
    System.int index  
)
```

Parameters

index

Index of the item to remove from the collection.

Return Value

A reference to the item that was removed from the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference




[XLColumnCollection Class](#)



[XLColumnCollection Members](#)

Properties

For a list of all members of this type, see [XLColumnCollection members](#).

Public Properties

	Name	Description
	Book	Gets a reference to the parent C1XLBook object.
	Count	Gets the number of items in the collection.
	Frozen	Gets or sets the number of frozen columns in the collection.

 Item	Returns a reference to the XLColumn object at the specified index.
 Sheet	Gets a reference to the parent XLSheet object.

[Top](#)

See Also

Reference

[XLColumnCollection Class](#)

[C1.C1Excel Namespace](#)

Book Property

Gets a reference to the parent [C1XLBook](#) object.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Book As C1XLBook</code>	
C#	
<code>public C1XLBook Book {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)

[XLColumnCollection Members](#)

Count Property

Gets the number of items in the collection.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Count As System.Integer</code>	
C#	
<code>public System.int Count {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)

[XLColumnCollection Members](#)

Frozen Property

Gets or sets the number of frozen columns in the collection.

Syntax

Visual Basic (Declaration)	
<code>Public Property Frozen As System.Integer</code>	
C#	
<code>public System.int Frozen {get; set;}</code>	

Remarks

Frozen columns are displayed on the right side of the sheet and do not scroll horizontally. They are useful for displaying row headers.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)

[XLColumnCollection Members](#)

Item Property

Returns a reference to the [XLColumn](#) object at the specified index.

Syntax

Visual Basic (Declaration)	
<pre>Public ReadOnly Default Property Item(_ ByVal index As System.Integer _) As XLColumn</pre>	
C#	
<pre>public XLColumn this[System.int index]; {get;}</pre>	

Parameters

index

Remarks

The indexer will create a new [XLColumn](#) object at the specified position if necessary. It never returns null.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)
[XLColumnCollection Members](#)

Sheet Property

Gets a reference to the parent [XLSheet](#) object.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property Sheet As XLSheet
```

C#

```
public XLSheet Sheet {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLColumnCollection Class](#)
[XLColumnCollection Members](#)

XLComment

Represents a comment associated with a cell.

Object Model

XLComment

Syntax

Visual Basic (Declaration)

```
Public Class XLComment
```

C#

```
public class XLComment
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLComment

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLComment Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents a comment associated with a cell.

Object Model

XLComment

Syntax

Visual Basic (Declaration)

```
Public Class XLComment
```

C#

```
public class XLComment
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLComment

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLComment Members](#)


[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)







The following tables list the members exposed by [XLComment](#).


Public Constructors

	Name	Description
	XLComment Constructor	Creates an instance of a XLComment object containing a specified range.

[Top](#)

Public Properties

	Name	Description
	Author	Gets a author for this comment.
	Cell	Gets a reference to the cell for this comment.
	ColumnIndex	Gets or sets the column index of the comment.
	IsShow	Gets or sets the show flag of the comment.
	RowIndex	Gets or sets the row index of the comment.
	Sheet	Gets a reference to the parent XLSheet object.

	TextBox	Gets a reference to the XLCommentShape object with context.
---	-------------------------	---

[Top](#)

Public Methods

	Name	Description
	Clone	Creates a new XLComment object that is a copy of the current instance.

[Top](#)

See Also

Reference

[XLComment Class](#)

[C1.C1Excel Namespace](#)

XLComment Constructor

Row that the comment applies to.

Column that the comment applies to.

Comment author.

Comment content.

Creates an instance of a [XLComment](#) object containing a specified range.

Syntax

Visual Basic (Declaration)	
<pre>Public Function New(_ ByVal row As System.Integer, _ ByVal col As System.Integer, _ ByVal author As System.String, _ ByVal text As System.String _)</pre>	
C#	

```
public XLComment(
    System.int row,
    System.int col,
    System.string author,
    System.string text
)
```

Parameters

row

Row that the comment applies to.

col

Column that the comment applies to.

author

Comment author.

text

Comment content.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also


Reference

[XLComment Class](#)

[XLComment Members](#)

Methods

>

Name	Description
 Clone	Creates a new XLComment object that is a copy of the current instance.

[Top](#)

See Also

Reference

[XLComment Class](#)

[C1.C1Excel Namespace](#)

Clone Method

Creates a new [XLComment](#) object that is a copy of the current instance.

Syntax

Visual Basic (Declaration)	
Public Function Clone() As XLComment	
C#	
public XLComment Clone()	

Return Value

A new [XLComment](#) object that is a copy of the current instance.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLComment Class](#)








[XLComment Members](#)

Properties

For a list of all members of this type, see [XLComment members](#).

Public Properties

Name	Description
------	-------------

	Author	Gets a author for this comment.
	Cell	Gets a reference to the cell for this comment.
	ColumnIndex	Gets or sets the column index of the comment.
	IsShow	Gets or sets the show flag of the comment.
	RowIndex	Gets or sets the row index of the comment.
	Sheet	Gets a reference to the parent XLSheet object.
	TextBox	Gets a reference to the XLCommentShape object with context.

[Top](#)

See Also

Reference

[XLComment Class](#)

[C1.C1Excel Namespace](#)

Author Property

Gets a author for this comment.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property Author As System.String
```

C#

```
public System.string Author {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLComment Class](#)

[XLComment Members](#)

Cell Property

Gets a reference to the cell for this comment.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Cell As XLCell</code>	
C#	
<code>public XLCell Cell {get;}</code>	

Remarks

A reference to the [XLCell](#) object at the comment coordinates, or null if there is no cell at the specified position.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLComment Class](#)

[XLComment Members](#)

ColumnIndex Property

Gets or sets the column index of the comment.

Syntax

Visual Basic (Declaration)	
<code>Public Property ColumnIndex As System.Integer</code>	
C#	
<code>public System.int ColumnIndex {get; set;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLComment Class](#)

[XLComment Members](#)

IsShow Property

Gets or sets the show flag of the comment.

Syntax

Visual Basic (Declaration)	
<code>Public Property IsShow As System.Boolean</code>	
C#	
<code>public System.bool IsShow {get; set;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLComment Class](#)

[XLComment Members](#)

RowIndex Property

Gets or sets the row index of the comment.

Syntax

Visual Basic (Declaration)

```
Public Property RowIndex As System.Integer
```

C#

```
public System.int RowIndex {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLComment Class](#)

[XLComment Members](#)

Sheet Property

Gets a reference to the parent [XLSheet](#) object.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property Sheet As XLSheet
```

C#

```
public XLSheet Sheet {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLComment Class](#)

[XLComment Members](#)

TextBox Property

Gets a reference to the [XLCommentShape](#) object with context.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property TextBox As XLCommentShape</code>	
C#	
<code>public XLCommentShape TextBox {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLComment Class](#)

[XLComment Members](#)

XLCommentCollection

Represents a collection of [XLComment](#) objects in a [XLSheet](#).

Object Model

XLCommentCollection

Syntax

Visual Basic (Declaration)	
<pre><System.Reflection.DefaultMemberAttribute("Item")> Public Class XLCommentCollection</pre>	
C#	
<pre>[System.Reflection.DefaultMemberAttribute("Item")] public class XLCommentCollection</pre>	

Inheritance Hierarchy

System.Object

C1.C1Excel.XLCommentCollection

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents a collection of [XLComment](#) objects in a [XLSheet](#).

Object Model

XLCommentCollection

Syntax

Visual Basic (Declaration)	
<pre><System.Reflection.DefaultMemberAttribute("Item")></pre>	

```
Public Class XLCommentCollection
```

```
C#
```

```
[System.Reflection.DefaultMemberAttribute("Item")]
```

```
public class XLCommentCollection
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLCommentCollection

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Members](#)




[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)

The following tables list the members exposed by [XLCommentCollection](#).

Public Properties

	Name	Description
	Count	Gets the number of XLComment objects in the collection.
	Item	Gets the XLComment object at the specified position in the collection.
	Sheet	Gets the XLSheet object that owns the collection.

[Top](#)

Public Methods

	Name	Description
≡	Add	Overloaded. Appends an XLComment object to the collection.
≡	Clear	Removes all XLComment objects from the collection.
≡	Contains	Checks whether the collection contains a specific XLComment object.
≡	IndexOf	Gets the index of a specific XLComment object in the collection.
≡	Insert	Inserts an XLComment object at a specific position in the collection.
≡	Remove	Removes an XLComment object from the collection.
≡	RemoveAt	Removes an XLComment object at a specific position from the collection.

[Top](#)

See Also

Reference

[XLCommentCollection Class](#)

[C1.C1Excel Namespace](#)

Methods

>

Name	Description
≡ Add	Overloaded. Appends an XLComment object to the collection.
≡ Clear	Removes all XLComment objects from the collection.
≡ Contains	Checks whether the collection contains a specific XLComment object.
≡ IndexOf	Gets the index of a specific XLComment object in the collection.
≡ Insert	Inserts an XLComment object at a specific position in the collection.
≡ Remove	Removes an XLComment object from the collection.

 [RemoveAt](#)

Removes an [XLComment](#) object at a specific position from the collection.

[Top](#)

See Also

Reference

[XLCommentCollection Class](#)

[C1.C1Excel Namespace](#)

Add Method

Appends an [XLComment](#) object to the collection.

Overload List

Overload	Description
Add(XLComment)	Appends an XLComment object to the collection.
Add(Int32,Int32,String)	Creates an XLComment object and appends it to the collection.
Add(Int32,Int32,String,String)	Creates an XLComment object and appends it to the collection.
Add(Int32,Int32,String,String,Boolean)	Creates an XLComment object and appends it to the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Class](#)

[XLCommentCollection Members](#)

Add(XLComment) Method

The [XLComment](#) object to add to the collection.

Appends an [XLComment](#) object to the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function Add(_ ByVal comment As XLComment _) As XLComment</pre>	
C#	
<pre>public XLComment Add(XLComment comment)</pre>	

Parameters

comment

The [XLComment](#) object to add to the collection.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added to the collection (usually because it overlaps another comment already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Class](#)

[XLCommentCollection Members](#)

[Overload List](#)

Add(Int32,Int32,String) Method

Index of the top row in the comment.

Index of the left column in the comment.

The author of the comment.

Creates an [XLComment](#) object and appends it to the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function Add(_ ByVal rowIndex As System.Integer, _ ByVal colIndex As System.Integer, _ ByVal author As System.String _) As XLComment</pre>	
C#	
<pre>public XLComment Add(System.int rowIndex, System.int colIndex, System.string author)</pre>	

Parameters

rowIndex

Index of the top row in the comment.

colIndex

Index of the left column in the comment.

author

The author of the comment.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added to the collection (usually because it overlaps another comment already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Class](#)
[XLCommentCollection Members](#)
[Overload List](#)

Add(Int32,Int32,String,String) Method

Index of the top row in the comment.

Index of the left column in the comment.

The author of the comment.

The context of the comment.

Creates an [XLComment](#) object and appends it to the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function Add(_ ByVal rowIndex As System.Integer, _ ByVal colIndex As System.Integer, _ ByVal author As System.String, _ ByVal text As System.String _) As XLComment</pre>	
C#	
<pre>public XLComment Add(System.int rowIndex, System.int colIndex, System.string author, System.string text</pre>	

)

Parameters

rowIndex

Index of the top row in the comment.

colIndex

Index of the left column in the comment.

author

The author of the comment.

text

The context of the comment.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added to the collection (usually because it overlaps another comment already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Class](#)

[XLCommentCollection Members](#)

[Overload List](#)

Add(Int32,Int32,String,String,Boolean) Method

Index of the top row in the comment.

Index of the left column in the comment.

The author of the comment.

The text of the comment.

Flag indicating whether to convert [author](#) and [text](#) to RTF format.

Creates an [XLComment](#) object and appends it to the collection.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Add( _  
    ByVal rowIndex As System.Integer, _  
    ByVal colIndex As System.Integer, _  
    ByVal author As System.String, _  
    ByVal text As System.String, _  
    ByVal toRtf As System.Boolean _  
) As XLComment
```

C#

```
public XLComment Add(  
    System.int rowIndex,  
    System.int colIndex,  
    System.string author,  
    System.string text,  
    System.bool toRtf  
)
```

Parameters

rowIndex

Index of the top row in the comment.

colIndex

Index of the left column in the comment.

author

The author of the comment.

text

The text of the comment.

toRtf

Flag indicating whether to convert [author](#) and [text](#) to RTF format.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added to the collection (usually because it overlaps another comment already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Class](#)
[XLCommentCollection Members](#)
[Overload List](#)

Clear Method

Removes all [XLComment](#) objects from the collection.

Syntax

Visual Basic (Declaration)

```
Public Sub Clear()
```

C#

```
public void Clear()
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Class](#)

[XLCommentCollection Members](#)

Contains Method

The [XLComment](#) object to look for.

Checks whether the collection contains a specific [XLComment](#) object.

Syntax

Visual Basic (Declaration)	
<pre>Public Function Contains(_ ByVal comment As XLComment _) As System.Boolean</pre>	
C#	
<pre>public System.bool Contains(XLComment comment)</pre>	

Parameters

comment

The [XLComment](#) object to look for.

Return Value

True if the collection contains the comment, false otherwise.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Class](#)

[XLCommentCollection Members](#)

IndexOf Method

The [XLComment](#) object to look for.

Gets the index of a specific [XLComment](#) object in the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Function IndexOf(_ ByVal comment As XLComment _) As System.Integer</pre>	
C#	
<pre>public System.int IndexOf(XLComment comment)</pre>	

Parameters

comment

The [XLComment](#) object to look for.

Return Value

The position of the object in the collection, or -1 if the object is not a member of the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Class](#)

[XLCommentCollection Members](#)

Insert Method

Position where the object will be inserted.

Object to insert in the collection.

Inserts an [XLComment](#) object at a specific position in the collection.

Syntax

Visual Basic (Declaration)

```
Public Function Insert( _  
    ByVal index As System.Integer, _  
    ByVal comment As XLComment _  
) As XLComment
```

C#

```
public XLComment Insert(  
    System.int index,  
    XLComment comment  
)
```

Parameters

index

Position where the object will be inserted.

comment

Object to insert in the collection.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added to the collection (usually because it overlaps another comment already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Class](#)
[XLCommentCollection Members](#)

Remove Method

The [XLComment](#) object to remove from the collection.

Removes an [XLComment](#) object from the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Sub Remove(_ ByVal comment As XLComment _)</pre>	
C#	
<pre>public void Remove(XLComment comment)</pre>	

Parameters

comment

The [XLComment](#) object to remove from the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Class](#)
[XLCommentCollection Members](#)

RemoveAt Method

Index of the object to remove from the collection.

Removes an [XLComment](#) object at a specific position from the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Sub RemoveAt(_ ByVal index As System.Integer _)</pre>	
C#	
<pre>public void RemoveAt(System.int index)</pre>	

Parameters

index

Index of the object to remove from the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference


[XLCommentCollection Class](#)



[XLCommentCollection Members](#)

Properties

For a list of all members of this type, see [XLCommentCollection members](#).

Public Properties

	Name	Description
	Count	Gets the number of XLComment objects in the collection.

 Item	Gets the XLComment object at the specified position in the collection.
 Sheet	Gets the XLSheet object that owns the collection.

[Top](#)

See Also

Reference

[XLCommentCollection Class](#)

[C1.C1Excel Namespace](#)

Count Property

Gets the number of [XLComment](#) objects in the collection.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Count As System.Integer</code>	
C#	
<code>public System.int Count {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Class](#)

[XLCommentCollection Members](#)

Item Property

Gets the [XLComment](#) object at the specified position in the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public ReadOnly Default Property Item(_ ByVal index As System.Integer _) As XLComment</pre>	
C#	
<pre>public XLComment this[System.int index]; {get;}</pre>	

Parameters

index

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Class](#)

[XLCommentCollection Members](#)

Sheet Property

Gets the [XLSheet](#) object that owns the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public ReadOnly Property Sheet As XLSheet</pre>	
C#	
<pre>public XLSheet Sheet {get;}</pre>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentCollection Class](#)

[XLCommentCollection Members](#)

XLCommentShape

Represents a text shape embedded in an [XLSheet](#).

Object Model

XLCommentShape

Syntax

Visual Basic (Declaration)

```
Public Class XLCommentShape  
    Inherits XLTextShape
```

C#

```
public class XLCommentShape : XLTextShape
```

Inheritance Hierarchy

System.Object

[C1.C1Excel.XLShape](#)

[C1.C1Excel.XLTextShape](#)

C1.C1Excel.XLCommentShape

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentShape Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents a text shape embedded in an [XLSheet](#).

Object Model

XLCommentShape

Syntax

Visual Basic (Declaration)

```
Public Class XLCommentShape
    Inherits XLTextShape
```

C#

```
public class XLCommentShape : XLTextShape
```

Inheritance Hierarchy

System.Object

[C1.C1Excel.XLShape](#)

[C1.C1Excel.XLTextShape](#)

C1.C1Excel.XLCommentShape

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentShape Members](#)


[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)






The following tables list the members exposed by [XLCommentShape](#).


Public Constructors










	Name	Description
	XLCommentShape Constructor	Overloaded.

[Top](#)

Public Properties


	Name	Description
	Bidirectional	Gets or sets a bidirectional count of text shape of this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	BottomMargin	Gets or sets the bottom margin of this XLTextShape , in twips. (Inherited from C1.C1Excel.XLTextShape)
	Column	Gets the index of the column to which the shape is attached. (Inherited from C1.C1Excel.XLShape)
	DashedLineStyle	Gets or sets the dash style of the line or border around the shape. (Inherited from C1.C1Excel.XLShape)
	HorizAlign	Gets or sets a horizontal alignment of the text in this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	Hyperlink	Gets or sets the hyperlink associated with the shape. (Inherited from C1.C1Excel.XLShape)
	Id	Gets the unique identifier of the shape. (Inherited from C1.C1Excel.XLShape)

 IsEmpty	Overridden. Determines whether the shape is empty.
 LeftMargin	Gets or sets the left margin area of this XLTextShape , in twips. (Inherited from C1.C1Excel.XLTextShape)
 LineColor	Gets or sets the color of the border around the shape. (Inherited from C1.C1Excel.XLShape)
 LineStyle	Gets or sets the style of the line or border around the shape. (Inherited from C1.C1Excel.XLShape)
 LineWidth	Gets or sets the width of the border around the shape, in twips. (Inherited from C1.C1Excel.XLShape)
 Locked	Gets or sets a locked of this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
 Orientation	Gets or sets a orientation of this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
 Rectangle	Gets or sets the rectangle that contains the shape, in twips. (Inherited from C1.C1Excel.XLShape)
 RightMargin	Gets or sets the right margin of this XLTextShape , in twips. (Inherited from C1.C1Excel.XLTextShape)
 Rotation	Gets or sets the rotation of the shape, in degrees. (Inherited from C1.C1Excel.XLShape)
 Row	Gets the index of the row to which the shape is attached. (Inherited from C1.C1Excel.XLShape)
 Sheet	Gets a reference to the parent XLSheet object. (Inherited from C1.C1Excel.XLShape)

	Text	Gets or sets a reference to the text string contained in this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	TextId	Gets or sets a text identifier of this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	TextScale	Gets or sets a text scale of this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	TextToFit	Gets or sets a fit to shape of the text flag of this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	TopMargin	Gets or sets the top margin of this XLTextShape , in twips. (Inherited from C1.C1Excel.XLTextShape)
	VertAlign	Gets or sets a vertical alignment of the text in this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	Visible	Determines whether the shape is visible.
	Workbook	Gets a reference to the parent C1XLBook object. (Inherited from C1.C1Excel.XLShape)
	Wrapped	Gets or sets a wrapped text of this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)

[Top](#)

Public Methods

	Name	Description
	Clone	Creates a new XLShape object that is a copy of the current instance. (Inherited from C1.C1Excel.XLShape)

[Top](#)

See Also

Reference

[XLCommentShape Class](#)

[C1.C1Excel Namespace](#)

XLCommentShape Constructor

Overload List

Overload	Description
XLCommentShape Constructor(String,Int32,Int32)	Initializes a new instance of an XLCommentShape .
XLCommentShape Constructor(String,Int32,Int32,Int32,Int32)	Initializes a new instance of an XLCommentShape .
XLCommentShape Constructor(String,Rectangle)	Initializes a new instance of an XLCommentShape .
XLCommentShape Constructor(String)	Initializes a new instance of an XLCommentShape .

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentShape Class](#)

[XLCommentShape Members](#)

[XLCommentShape Constructor\(String,Int32,Int32\)](#)

The text contained in the new [XLCommentShape](#).

The horizontal position of the new shape with respect to the cell, in twips.

The vertical position of the new shape with respect to the cell, in twips.

Initializes a new instance of an [XLCommentShape](#).

Syntax

Visual Basic (Declaration)	
<pre>Public Function New(_ ByVal text As System.String, _ ByVal x As System.Integer, _ ByVal y As System.Integer _)</pre>	
C#	
<pre>public XLCommentShape(System.string text, System.int x, System.int y)</pre>	

Parameters

text

The text contained in the new [XLCommentShape](#).

x

The horizontal position of the new shape with respect to the cell, in twips.

y

The vertical position of the new shape with respect to the cell, in twips.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentShape Class](#)
[XLCommentShape Members](#)
[Overload List](#)

XLCommentShape Constructor(String,Int32,Int32,Int32,Int32)

The text contained in the new [XLCommentShape](#).

The horizontal position of the image with respect to the cell, in twips.

The vertical position of the image with respect to the cell, in twips.

The width of the image, in twips.

The height of the image, in twips.

Initializes a new instance of an [XLCommentShape](#).

Syntax

Visual Basic (Declaration)

```
Public Function New( _  
    ByVal text As System.String, _  
    ByVal x As System.Integer, _  
    ByVal y As System.Integer, _  
    ByVal width As System.Integer, _  
    ByVal height As System.Integer _  
)
```

C#

```
public XLCommentShape(  
    System.string text,  
    System.int x,  
    System.int y,  
    System.int width,  
    System.int height  
)
```

Parameters

text

The text contained in the new [XLCommentShape](#).

x

The horizontal position of the image with respect to the cell, in twips.

y

The vertical position of the image with respect to the cell, in twips.

width

The width of the image, in twips.

height

The height of the image, in twips.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentShape Class](#)
[XLCommentShape Members](#)
[Overload List](#)

XLCommentShape Constructor(String,Rectangle)

The text contained in the new [XLCommentShape](#).

The rectangle that specifies the image size and position with respect to the cell, in twips.

Initializes a new instance of an [XLCommentShape](#).

Syntax

Visual Basic (Declaration)

```
Public Function New( _  
    ByVal text As System.String, _  
    ByVal rc As System.Drawing.Rectangle _
```

```
)
```

C#

```
public XLCommentShape(  
    System.string text,  
    System.Drawing.Rectangle rc  
)
```

Parameters

text

The text contained in the new [XLCommentShape](#).

rc

The rectangle that specifies the image size and position with respect to the cell, in twips.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentShape Class](#)
[XLCommentShape Members](#)
[Overload List](#)

XLCommentShape Constructor(String)

The text contained in the new [XLCommentShape](#).

Initializes a new instance of an [XLCommentShape](#).

Syntax

Visual Basic (Declaration)

```
Public Function New( _  
    ByVal text As System.String _  
)
```

C#

```
public XLCommentShape(  
    System.string text  
)
```

Parameters

text

The text contained in the new [XLCommentShape](#).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also




Reference














[XLCommentShape Class](#)
[XLCommentShape Members](#)
[Overload List](#)













Properties

For a list of all members of this type, see [XLCommentShape members](#).

Public Properties

	Name	Description
	Bidirectional	Gets or sets a bidirectional count of text shape of this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	BottomMargin	Gets or sets the bottom margin of this XLTextShape , in twips. (Inherited from C1.C1Excel.XLTextShape)
	Column	Gets the index of the column to which the shape is attached. (Inherited from C1.C1Excel.XLShape)

	DashedLineStyle	Gets or sets the dash style of the line or border around the shape. (Inherited from C1.C1Excel.XLShape)
	HorizAlign	Gets or sets a horizontal alignment of the text in this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	Hyperlink	Gets or sets the hyperlink associated with the shape. (Inherited from C1.C1Excel.XLShape)
	Id	Gets the unique identifier of the shape. (Inherited from C1.C1Excel.XLShape)
	IsEmpty	Overridden. Determines whether the shape is empty.
	LeftMargin	Gets or sets the left margin area of this XLTextShape , in twips. (Inherited from C1.C1Excel.XLTextShape)
	LineColor	Gets or sets the color of the border around the shape. (Inherited from C1.C1Excel.XLShape)
	LineStyle	Gets or sets the style of the line or border around the shape. (Inherited from C1.C1Excel.XLShape)
	LineWidth	Gets or sets the width of the border around the shape, in twips. (Inherited from C1.C1Excel.XLShape)
	Locked	Gets or sets a locked of this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	Orientation	Gets or sets a orientation of this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	Rectangle	Gets or sets the rectangle that contains the shape, in twips. (Inherited from C1.C1Excel.XLShape)
	RightMargin	Gets or sets the right margin of this XLTextShape , in twips. (Inherited from

		C1.C1Excel.XLTextShape)
	Rotation	Gets or sets the rotation of the shape, in degrees. (Inherited from C1.C1Excel.XLShape)
	Row	Gets the index of the row to which the shape is attached. (Inherited from C1.C1Excel.XLShape)
	Sheet	Gets a reference to the parent XLSheet object. (Inherited from C1.C1Excel.XLShape)
	Text	Gets or sets a reference to the text string contained in this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	TextId	Gets or sets a text identifier of this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	TextScale	Gets or sets a text scale of this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	TextToFit	Gets or sets a fit to shape of the text flag of this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	TopMargin	Gets or sets the top margin of this XLTextShape , in twips. (Inherited from C1.C1Excel.XLTextShape)
	VertAlign	Gets or sets a vertical alignment of the text in this XLTextShape . (Inherited from C1.C1Excel.XLTextShape)
	Visible	Determines whether the shape is visible.
	Workbook	Gets a reference to the parent C1XLBook object. (Inherited from C1.C1Excel.XLShape)
	Wrapped	Gets or sets a wrapped text of this XLTextShape . (Inherited from

		C1.C1Excel.XLTextShape)
--	--	--

[Top](#)

See Also

Reference

[XLCommentShape Class](#)

[C1.C1Excel Namespace](#)

IsEmpty Property

Determines whether the shape is empty.

Syntax

Visual Basic (Declaration)	
<code>Public Overrides ReadOnly Property IsEmpty As System.Boolean</code>	
C#	
<code>public override System.bool IsEmpty {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentShape Class](#)

[XLCommentShape Members](#)

Visible Property

Determines whether the shape is visible.

Syntax

Visual Basic (Declaration)	
----------------------------	--

Public Property Visible **As** System.Boolean

C#

```
public System.bool Visible {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLCommentShape Class](#)

[XLCommentShape Members](#)

XLNamedRange

Represents a named range of [XLCell](#) objects on one or several worksheets.

Object Model

XLNamedRange

Syntax

Visual Basic (Declaration)

```
Public Class XLNamedRange
```

C#

```
public class XLNamedRange
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLNamedRange

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRange Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents a named range of [XLCell](#) objects on one or several worksheets.

Object Model

XLNamedRange

Syntax

Visual Basic (Declaration)

```
Public Class XLNamedRange
```

C#

```
public class XLNamedRange
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLNamedRange

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

XLNamedRange Members









C1.C1Excel Namespace

Members

[Properties](#) [Methods](#)


The following tables list the members exposed by [XLNamedRange](#).


Public Properties

	Name	Description
	Book	Gets a reference to the parent C1XLBook object.
	CellRange	Gets a reference to the XLCellRange object that determines which cells are contained in the current named range.
	CellRanges	Gets a reference to array of the XLCellRange objects that determines which cells are contained in the current named range.
	Comment	Gets or sets the comment text associated with the current named range.
	IsBuiltInName	Determines whether the range has a built-in name.
	IsEmpty	Determines whether the current range is empty.
	IsNameOnly	Determines whether the current named range has a valid name but does not define a range.
	Name	Gets or sets the name of the current named range.

[Top](#)

Public Methods

	Name	Description
	Clone	Creates a new XLNamedRange object that is a copy of the current instance.

 Contains	Determines whether the range contains a specific cell.
--	--

[Top](#)



See Also

Reference

[XLNamedRange Class](#)
[C1.C1Excel Namespace](#)

Methods

>

Name	Description
 Clone	Creates a new XLNamedRange object that is a copy of the current instance.
 Contains	Determines whether the range contains a specific cell.

[Top](#)

See Also

Reference

[XLNamedRange Class](#)
[C1.C1Excel Namespace](#)

Clone Method

Creates a new [XLNamedRange](#) object that is a copy of the current instance.

Syntax

Visual Basic (Declaration)	
Public Function Clone() As XLNamedRange	
C#	
public XLNamedRange Clone()	

Return Value

A new [XLNamedRange](#) object that is a copy of the current instance.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRange Class](#)

[XLNamedRange Members](#)

Contains Method

The worksheet containing the cell.

The row index of the cell.

The column index of the cell.

Determines whether the range contains a specific cell.

Syntax

Visual Basic (Declaration)	
<pre>Public Function Contains(_ ByVal sheet As XLSheet, _ ByVal row As System.Integer, _ ByVal col As System.Integer _) As System.Boolean</pre>	
C#	
<pre>public System.bool Contains(XLSheet sheet, System.int row, System.int col)</pre>	

Parameters

sheet

The worksheet containing the cell.

row

The row index of the cell.

col

The column index of the cell.

Return Value

True if the current named range contains the specified cell, false otherwise.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference





[XLNamedRange Class](#)





[XLNamedRange Members](#)

Properties

For a list of all members of this type, see [XLNamedRange members](#).

Public Properties

	Name	Description
	Book	Gets a reference to the parent C1XLBook object.
	CellRange	Gets a reference to the XLCellRange object that determines which cells are contained in the current named range.
	CellRanges	Gets a reference to array of the XLCellRange objects that determines which cells are contained in the current named range.
	Comment	Gets or sets the comment text associated with the current named range.

	IsBuiltInName	Determines whether the range has a built-in name.
	IsEmpty	Determines whether the current range is empty.
	IsNameOnly	Determines whether the current named range has a valid name but does not define a range.
	Name	Gets or sets the name of the current named range.

[Top](#)

See Also

Reference

[XLNamedRange Class](#)

[C1.C1Excel Namespace](#)

Book Property

Gets a reference to the parent [C1XLBook](#) object.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property Book As C1XLBook
```

C#

```
public C1XLBook Book {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRange Class](#)

[XLNamedRange Members](#)

CellRange Property

Gets a reference to the [XLCellRange](#) object that determines which cells are contained in the current named range.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property CellRange As XLCellRange
```

C#

```
public XLCellRange CellRange {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRange Class](#)

[XLNamedRange Members](#)

CellRanges Property

Gets a reference to array of the [XLCellRange](#) objects that determines which cells are contained in the current named range.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property CellRanges As XLCellRange()
```

C#

```
public XLCellRange[] CellRanges {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRange Class](#)

[XLNamedRange Members](#)

Comment Property

Gets or sets the comment text associated with the current named range.

Syntax

Visual Basic (Declaration)	
Public Property Comment As System.String	
C#	
public System.string Comment { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRange Class](#)

[XLNamedRange Members](#)

IsBuiltInName Property

Determines whether the range has a built-in name.

Syntax

Visual Basic (Declaration)	
----------------------------	--

<code>Public ReadOnly Property IsBuiltInName As System.Boolean</code>	
C#	
<code>public System.bool IsBuiltInName {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRange Class](#)

[XLNamedRange Members](#)

IsEmpty Property

Determines whether the current range is empty.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property IsEmpty As System.Boolean</code>	
C#	
<code>public System.bool IsEmpty {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRange Class](#)

[XLNamedRange Members](#)

IsNameOnly Property

Determines whether the current named range has a valid name but does not define a range.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property IsNameOnly As System.Boolean
```

C#

```
public System.bool IsNameOnly {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRange Class](#)

[XLNamedRange Members](#)

Name Property

Gets or sets the name of the current named range.

Syntax

Visual Basic (Declaration)

```
Public Property Name As System.String
```

C#

```
public System.string Name {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRange Class](#)

[XLNamedRange Members](#)

XLNamedRangeCollection

Represents a collection of [XLNamedRange](#) objects.

Object Model

XLNamedRangeCollection

Syntax

Visual Basic (Declaration)

```
<System.Reflection.DefaultMemberAttribute("Item")>  
Public Class XLNamedRangeCollection
```

C#

```
[System.Reflection.DefaultMemberAttribute("Item")]  
public class XLNamedRangeCollection
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLNamedRangeCollection

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents a collection of [XLNamedRange](#) objects.

Object Model

[XLNamedRangeCollection](#)

Syntax

Visual Basic (Declaration)	
<pre><System.Reflection.DefaultMemberAttribute("Item")> Public Class XLNamedRangeCollection</pre>	
C#	
<pre>[System.Reflection.DefaultMemberAttribute("Item")] public class XLNamedRangeCollection</pre>	

Inheritance Hierarchy

System.Object

C1.C1Excel.XLNamedRangeCollection

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Members](#)




[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)








The following tables list the members exposed by [XLNamedRangeCollection](#).

Public Properties

	Name	Description
	Book	Gets a reference to the parent C1XLBook object.
	Count	Gets the number of XLNamedRange objects in the collection.
	Item	Overloaded. Gets a reference to the XLNamedRange object at the specified index. Returns null if an invalid index is specified.

[Top](#)

Public Methods

	Name	Description
	Add	Overloaded. Appends an XLNamedRange object to the collection.
	Clear	Removes all XLNamedRange objects from the collection.
	Contains	Overloaded. Checks whether the collection contains a specific XLNamedRange object.
	IndexOf	Overloaded. Gets the position of an XLNamedRange object in the collection.
	Insert	Inserts an XLNamedRange object at a specific position in the collection.
	Remove	Overloaded. Removes an XLNamedRange object from the collection.
	RemoveAt	Removes an XLNamedRange object at a specific position from the collection.

[Top](#)

See Also








Reference

XLNamedRangeCollection Class

C1.C1Excel Namespace

Methods

>

Name	Description
 Add	Overloaded. Appends an XLNamedRange object to the collection.
 Clear	Removes all XLNamedRange objects from the collection.
 Contains	Overloaded. Checks whether the collection contains a specific XLNamedRange object.
 IndexOf	Overloaded. Gets the position of an XLNamedRange object in the collection.
 Insert	Inserts an XLNamedRange object at a specific position in the collection.
 Remove	Overloaded. Removes an XLNamedRange object from the collection.
 RemoveAt	Removes an XLNamedRange object at a specific position from the collection.

[Top](#)

See Also

Reference

XLNamedRangeCollection Class

C1.C1Excel Namespace

Add Method

Appends an [XLNamedRange](#) object to the collection.

Overload List

Overload	Description
Add(XLNamedRange)	Appends an XLNamedRange object to the collection.

Add(String,XLCellRange)	Creates an XLNamedRange object and appends it to the collection.
Add(String,XLCellRange[])	Creates an XLNamedRange object and appends it to the collection.
Add(String,XLSheet,Int32,Int32)	Creates an XLNamedRange object and appends it to the collection. This overload creates a range containing a single cell.
Add(String,Int32,Int32,Int32,Int32)	Creates an XLNamedRange object and appends it to the collection. This overload allows you to create a 3-D reference to the same range of cells on multiple worksheets.
Add(String,XLSheet,Int32,Int32,Int32,Int32)	Creates an XLNamedRange object and appends it to the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)

[XLNamedRangeCollection Members](#)

Add(XLNamedRange) Method

The [XLNamedRange](#) object to add to the collection.

Appends an [XLNamedRange](#) object to the collection.

Syntax

Visual Basic (Declaration)


```
Public Overloads Function Add( _
    ByVal namedRange As XLNamedRange _
) As XLNamedRange
```

C#

```
public XLNamedRange Add(
    XLNamedRange namedRange
)
```

Parameters

namedRange

The [XLNamedRange](#) object to add to the collection.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added (usually because it overlaps another cell range already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)
[Overload List](#)

Add(String,XLCellRange) Method

The name of the named range to create.

The [XLCellRange](#) to include in the named range.

Creates an [XLNamedRange](#) object and appends it to the collection.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Add( _
    ByVal name As System.String, _
    ByVal cellRange As XLCellRange _
) As XLNamedRange
```

C#

```
public XLNamedRange Add(
    System.string name,
    XLCellRange cellRange
)
```

Parameters

name

The name of the named range to create.

cellRange

The [XLCellRange](#) to include in the named range.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added (usually because it overlaps another cell range already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)
[Overload List](#)

Add(String,XLCellRange[]) Method

The name of the named range to create.

The array of [XLCellRange](#) to include in the named range.

Creates an [XLNamedRange](#) object and appends it to the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function Add(_ ByVal name As System.String, _ ByVal cellRanges() As XLCellRange _) As XLNamedRange</pre>	
C#	
<pre>public XLNamedRange Add(System.string name, XLCellRange[] cellRanges)</pre>	

Parameters

name

The name of the named range to create.

cellRanges

The array of [XLCellRange](#) to include in the named range.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added (usually because it overlaps another cell range already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)
[Overload List](#)

Add(String,XLSheet,Int32,Int32) Method

The name of [XLNamedRange](#) object.

The worksheet the range belongs to.

The row index of the cell in the range.

The column index of the cell in the range.

Creates an [XLNamedRange](#) object and appends it to the collection. This overload creates a range containing a single cell.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Add( _  
    ByVal name As System.String, _  
    ByVal sheet As XLSheet, _  
    ByVal rowIndex As System.Integer, _  
    ByVal colIndex As System.Integer _  
) As XLNamedRange
```

C#

```
public XLNamedRange Add(  
    System.string name,  
    XLSheet sheet,  
    System.int rowIndex,  
    System.int colIndex  
)
```

Parameters

name

The name of [XLNamedRange](#) object.

sheet

The worksheet the range belongs to.

rowIndex

The row index of the cell in the range.

colIndex

The column index of the cell in the range.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added (usually because it overlaps another cell range already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)

[XLNamedRangeCollection Members](#)

[Overload List](#)

Add(String,Int32,Int32,Int32,Int32) Method

The name of [XLNamedRange](#) object.

The index of the first [XLSheet](#) to include.

The index of the last [XLSheet](#) to include.

The index of the top row in the cell range.

The index of the left column in the cell range.

Creates an [XLNamedRange](#) object and appends it to the collection. This overload allows you to create a 3-D reference to the same range of cells on multiple worksheets.

Syntax

Visual Basic (Declaration)	
----------------------------	--

```
Public Overloads Function Add( _
    ByVal name As System.String, _
    ByVal firstSheetIndex As System.Integer, _
    ByVal lastSheetIndex As System.Integer, _
    ByVal rowIndex As System.Integer, _
    ByVal colIndex As System.Integer _
) As XLNamedRange
```

C#

```
public XLNamedRange Add(
    System.string name,
    System.int firstSheetIndex,
    System.int lastSheetIndex,
    System.int rowIndex,
    System.int colIndex
)
```

Parameters

name

The name of [XLNamedRange](#) object.

firstSheetIndex

The index of the first [XLSheet](#) to include.

lastSheetIndex

The index of the last [XLSheet](#) to include.

rowIndex

The index of the top row in the cell range.

colIndex

The index of the left column in the cell range.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added (usually because it overlaps another cell range already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)
[Overload List](#)

Add(String,XLSheet,Int32,Int32,Int32,Int32) Method

The name of [XLNamedRange](#) object.

The worksheet containing the range.

The index of the top row in the cell range.

The index of the left column in the cell range.

The number of rows in the cell range.

The number of columns in the cell range.

Creates an [XLNamedRange](#) object and appends it to the collection.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Add( _  
    ByVal name As System.String, _  
    ByVal sheet As XLSheet, _  
    ByVal rowIndex As System.Integer, _  
    ByVal colIndex As System.Integer, _  
    ByVal rowCount As System.Integer, _  
    ByVal colCount As System.Integer _  
) As XLNamedRange
```

C#

```
public XLNamedRange Add(  
    System.string name,  
    XLSheet sheet,  
    System.int rowIndex,  
    System.int colIndex,  
    System.int rowCount,  
    System.int colCount  
)
```

Parameters

name

The name of `XLNamedRange` object.

sheet

The worksheet containing the range.

rowIndex

The index of the top row in the cell range.

colIndex

The index of the left column in the cell range.

rowCount

The number of rows in the cell range.

colCount

The number of columns in the cell range.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added (usually because it overlaps another cell range already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)
[Overload List](#)

Clear Method

Removes all [XLNamedRange](#) objects from the collection.

Syntax

Visual Basic (Declaration)	
Public Sub Clear()	
C#	
public void Clear()	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)

Contains Method

Checks whether the collection contains a specific [XLNamedRange](#) object.

Overload List

Overload	Description
Contains(XLNamedRange)	Checks whether the collection contains a specific XLNamedRange object.

Contains(String)	Checks whether the collection contains an XLNamedRange object with the specified name.
----------------------------------	--

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)

[XLNamedRangeCollection Members](#)

Contains(XLNamedRange) Method

The [XLNamedRange](#) object to look for.

Checks whether the collection contains a specific [XLNamedRange](#) object.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function Contains(_ ByVal namedRange As XLNamedRange _) As System.Boolean</pre>	
C#	
<pre>public System.bool Contains(XLNamedRange namedRange)</pre>	

Parameters

namedRange

The [XLNamedRange](#) object to look for.

Return Value

True if the collection contains the range, false otherwise.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)
[Overload List](#)

Contains(String) Method

The name of the [XLNamedRange](#) object to look for.

Checks whether the collection contains an [XLNamedRange](#) object with the specified name.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Contains( _  
    ByVal name As System.String _  
) As System.Boolean
```

C#

```
public System.bool Contains(  
    System.string name  
)
```

Parameters

name

The name of the [XLNamedRange](#) object to look for.

Return Value

True if the collection contains the range, false otherwise.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)
[Overload List](#)

IndexOf Method

Gets the position of an [XLNamedRange](#) object in the collection.

Overload List

Overload	Description
IndexOf(XLNamedRange)	Gets the position of an XLNamedRange object in the collection.
IndexOf(String)	Gets the position of an XLNamedRange object with the specified name in the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)

IndexOf(XLNamedRange) Method

The [XLNamedRange](#) object to look for.

Gets the position of an [XLNamedRange](#) object in the collection.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function IndexOf( _  
    ByVal nr As XLNamedRange _  
) As System.Integer
```

C#

```
public System.int IndexOf(  
    XLNamedRange nr  
)
```

Parameters

nr

The [XLNamedRange](#) object to look for.

Return Value

The position of the object in the collection, or -1 if the object is not a member of the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)
[Overload List](#)

IndexOf(String) Method

The name of the [XLNamedRange](#) object to look for.

Gets the position of an [XLNamedRange](#) object with the specified name in the collection.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function IndexOf( _  
    ByVal name As System.String _  
) As System.Integer
```

C#

```
public System.int IndexOf(  
    System.string name  
)
```

Parameters

name

The name of the [XLNamedRange](#) object to look for.

Return Value

The position of the object in the collection, or -1 if the object is not a member of the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)
[Overload List](#)

Insert Method

The position where the object will be inserted.

The named range to insert in the collection.

Inserts an [XLNamedRange](#) object at a specific position in the collection.

Syntax

Visual Basic (Declaration)

```
Public Function Insert( _  
    ByVal index As System.Integer, _  
    ByVal namedRange As XLNamedRange _  
) As XLNamedRange
```

C#

```
public XLNamedRange Insert(  
    System.int index,  
    XLNamedRange namedRange  
)
```

Parameters

index

The position where the object will be inserted.

namedRange

The named range to insert in the collection.

Return Value

A reference to the object if it was successfully added to the collection, or null if the object could not be added (usually because it overlaps another cell range already in the collection).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)

[XLNamedRangeCollection Members](#)

Remove Method

Removes an [XLNamedRange](#) object from the collection.

Overload List

Overload	Description
Remove(XLNamedRange)	Removes an XLNamedRange object from the collection.
Remove(String)	Removes an XLNamedRange object with the specified name from the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)

[XLNamedRangeCollection Members](#)

Remove(XLNamedRange) Method

The [XLNamedRange](#) object to remove from the collection.

Removes an [XLNamedRange](#) object from the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function Remove(_ ByVal nr As XLNamedRange _) As System.Boolean</pre>	
C#	
<pre>public System.bool Remove(XLNamedRange nr)</pre>	

Parameters

nr

The [XLNamedRange](#) object to remove from the collection.

Return Value

True if the object was removed, false if it was not a member of the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)
[Overload List](#)

Remove(String) Method

The name of the range to remove from the collection (case-insensitive).

Removes an [XLNamedRange](#) object with the specified name from the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Sub Remove(_ ByVal name As System.String _)</pre>	
C#	
<pre>public void Remove(System.string name)</pre>	

Parameters

name

The name of the range to remove from the collection (case-insensitive).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)
[Overload List](#)

RemoveAt Method

The index of the object to remove from the collection.

Removes an [XLNamedRange](#) object at a specific position from the collection.

Syntax

Visual Basic (Declaration)

```
Public Sub RemoveAt( _  
    ByVal index As System.Integer _  
)
```

C#

```
public void RemoveAt(  
    System.int index  
)
```

Parameters

index

The index of the object to remove from the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference




[XLNamedRangeCollection Class](#)

[XLNamedRangeCollection Members](#)

Properties

For a list of all members of this type, see [XLNamedRangeCollection members](#).

Public Properties

	Name	Description
	Book	Gets a reference to the parent C1XLBook object.
	Count	Gets the number of XLNamedRange objects in the collection.
	Item	Overloaded. Gets a reference to the XLNamedRange object at the specified index. Returns null if an invalid index is specified.

[Top](#)

See Also

Reference

[XLNamedRangeCollection Class](#)

[C1.C1Excel Namespace](#)

Book Property

Gets a reference to the parent [C1XLBook](#) object.

Syntax

Visual Basic (Declaration)	
Public ReadOnly Property Book As C1XLBook	
C#	
public C1XLBook Book { get ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)

[XLNamedRangeCollection Members](#)

Count Property

Gets the number of [XLNamedRange](#) objects in the collection.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Count As System.Integer</code>	
C#	
<code>public System.int Count {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)

[XLNamedRangeCollection Members](#)

Item Property

Gets a reference to the [XLNamedRange](#) object at the specified index. Returns null if an invalid index is specified.

Overload List

Overload	Description
Item(Int32)	Gets a reference to the XLNamedRange object at the specified index. Returns null if an invalid index is specified.
Item(String)	Gets a reference to the XLNamedRange object with the specified name. Returns null if an object with the specified name could not be found in the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)

[XLNamedRangeCollection Members](#)

Item(Int32) Property

Gets a reference to the [XLNamedRange](#) object at the specified index. Returns null if an invalid index is specified.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads ReadOnly Property Item(_ ByVal index As System.Integer _) As XLNamedRange</pre>	
C#	
<pre>public XLNamedRange Item(System.int index) {get;}</pre>	

Parameters

index

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)
[Overload List](#)

Item(String) Property

Gets a reference to the [XLNamedRange](#) object with the specified name. Returns null if an object with the specified name could not be found in the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads ReadOnly Property Item(_ ByVal name As System.String _) As XLNamedRange</pre>	
C#	
<pre>public XLNamedRange Item(System.string name) {get;}</pre>	

Parameters

name

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLNamedRangeCollection Class](#)
[XLNamedRangeCollection Members](#)
[Overload List](#)

XLOpaqueShape

Represents a shape that was inserted in the sheet using Excel and is preserved but not fully exposed by the [C1XLBook](#) component.

Object Model

XLOpaqueShape

Syntax

Visual Basic (Declaration)

```
Public Class XLOpaqueShape  
    Inherits XLShape
```

C#

```
public class XLOpaqueShape : XLShape
```

Remarks

C1ExcelBook can load and save all types of shapes present in Excel sheets.

However, only image shapes are fully exposed (as [XLPictureShape](#) objects).

All other object types are loaded and saved as **XLOpaqueShape** objects that cannot be modified. These include graphical elements (such as lines, rectangles, and arcs), VBA controls (such as edit boxes, and buttons), and comments.

Inheritance Hierarchy

System.Object

[C1.C1Excel.XLShape](#)

C1.C1Excel.XLOpaqueShape

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLOpaqueShape Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents a shape that was inserted in the sheet using Excel and is preserved but not fully exposed by the [C1XLBook](#) component.

Object Model

XLOpaqueShape

Syntax

Visual Basic (Declaration)

```
Public Class XLOpaqueShape
    Inherits XLShape
```

C#

```
public class XLOpaqueShape : XLShape
```

Remarks

C1ExcelBook can load and save all types of shapes present in Excel sheets.

However, only image shapes are fully exposed (as [XLPictureShape](#) objects).

All other object types are loaded and saved as **XLOpaqueShape** objects that cannot be modified. These include graphical elements (such as lines, rectangles, and arcs), VBA controls (such as edit boxes, and buttons), and comments.

Inheritance Hierarchy

System.Object

[C1.C1Excel.XLShape](#)

C1.C1Excel.XLOpaqueShape

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLOpaqueShape Members](#)







[C1.C1Excel Namespace](#)




Members

[Properties](#) [Methods](#)

The following tables list the members exposed by [XLOpaqueShape](#).


Public Properties

	Name	Description
	Column	Gets the index of the column to which the shape is attached. (Inherited from C1.C1Excel.XLShape)
	DashedLineStyle	Gets or sets the dash style of the line or border around the shape. (Inherited from C1.C1Excel.XLShape)
	Hyperlink	Gets or sets the hyperlink associated with the shape. (Inherited from C1.C1Excel.XLShape)
	Id	Gets the unique identifier of the shape. (Inherited from C1.C1Excel.XLShape)
	IsEmpty	Overridden. Determines whether the shape is empty.
	LineColor	Gets or sets the color of the border around the shape. (Inherited from

		C1.C1Excel.XLShape)
	LineStyle	Gets or sets the style of the line or border around the shape. (Inherited from C1.C1Excel.XLShape)
	LineWidth	Gets or sets the width of the border around the shape, in twips. (Inherited from C1.C1Excel.XLShape)
	Rectangle	Gets or sets the rectangle that contains the shape, in twips. (Inherited from C1.C1Excel.XLShape)
	Rotation	Gets or sets the rotation of the shape, in degrees. (Inherited from C1.C1Excel.XLShape)
	Row	Gets the index of the row to which the shape is attached. (Inherited from C1.C1Excel.XLShape)
	Sheet	Gets a reference to the parent XLSheet object. (Inherited from C1.C1Excel.XLShape)
	Workbook	Gets a reference to the parent C1XLBook object. (Inherited from C1.C1Excel.XLShape)

[Top](#)

Public Methods

	Name	Description
	Clone	Creates a new XLShape object that is a copy of the current instance. (Inherited from C1.C1Excel.XLShape)

[Top](#)











See Also




Reference

Properties

For a list of all members of this type, see [XLOpaqueShape members](#).

Public Properties

	Name	Description
	Column	Gets the index of the column to which the shape is attached. (Inherited from C1.C1Excel.XLShape)
	DashedLineStyle	Gets or sets the dash style of the line or border around the shape. (Inherited from C1.C1Excel.XLShape)
	Hyperlink	Gets or sets the hyperlink associated with the shape. (Inherited from C1.C1Excel.XLShape)
	Id	Gets the unique identifier of the shape. (Inherited from C1.C1Excel.XLShape)
	IsEmpty	Overridden. Determines whether the shape is empty.
	LineColor	Gets or sets the color of the border around the shape. (Inherited from C1.C1Excel.XLShape)
	LineStyle	Gets or sets the style of the line or border around the shape. (Inherited from C1.C1Excel.XLShape)
	LineWidth	Gets or sets the width of the border around the shape, in twips. (Inherited from C1.C1Excel.XLShape)
	Rectangle	Gets or sets the rectangle that contains the shape, in twips. (Inherited from C1.C1Excel.XLShape)
	Rotation	Gets or sets the rotation of the shape, in degrees. (Inherited from C1.C1Excel.XLShape)

	Row	Gets the index of the row to which the shape is attached. (Inherited from C1.C1Excel.XLShape)
	Sheet	Gets a reference to the parent XLSheet object. (Inherited from C1.C1Excel.XLShape)
	Workbook	Gets a reference to the parent C1XLBook object. (Inherited from C1.C1Excel.XLShape)

[Top](#)

See Also

Reference

[XLOpaqueShape Class](#)

[C1.C1Excel Namespace](#)

IsEmpty Property

Determines whether the shape is empty.

Syntax

Visual Basic (Declaration)	
<code>Public Overrides ReadOnly Property IsEmpty As System.Boolean</code>	
C#	
<code>public override System.bool IsEmpty {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLOpaqueShape Class](#)

[XLOpaqueShape Members](#)

XLPictureShape

Represents an [Image](#) embedded in an [XLSheet](#).

Object Model

XLPictureShape

Syntax

Visual Basic (Declaration)	
<pre>Public Class XLPictureShape Inherits XLShape</pre>	
C#	
<pre>public class XLPictureShape : XLShape</pre>	

Remarks

XLPictureShape derives from the generic [XLShape](#) class to expose properties of images embedded in sheets. These properties include the actual [Image](#) as well as information on how it should be displayed, including [Brightness](#), [Contrast](#), and clipping information.

You can add images to cells simply by assigning [Image](#) objects directly to the [Value](#) property of [XLCell](#) objects. In this case, [C1XLBook](#) will create and initialize an **XLPictureShape** automatically. However, this method does not provide a lot of flexibility in terms of aligning, scaling, and clipping the image.

A more flexible option is to create an **XLPictureShape** object in code, using the constructor that takes alignment and scaling parameters, and then assign this **XLPictureShape** object to a cell's [Value](#) property.

Inheritance Hierarchy

System.Object

[C1.C1Excel.XLShape](#)

C1.C1Excel.XLPictureShape

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents an [Image](#) embedded in an [XLSheet](#).

Object Model

XLPictureShape

Syntax

Visual Basic (Declaration)	
Public Class XLPictureShape Inherits XLShape	
C#	
public class XLPictureShape : XLShape	

Remarks

XLPictureShape derives from the generic [XLShape](#) class to expose properties of images embedded in sheets. These properties include the actual [Image](#) as well as information on how it should be displayed, including [Brightness](#), [Contrast](#), and clipping information.

You can add images to cells simply by assigning [Image](#) objects directly to the [Value](#) property of [XLCell](#) objects. In this case, [C1XLBook](#) will create and initialize an **XLPictureShape** automatically. However, this method does not provide a lot of flexibility in terms of aligning, scaling, and clipping the image.

A more flexible option is to create an **XLPictureShape** object in code, using the constructor that takes alignment and scaling parameters, and then assign this **XLPictureShape** object to a cell's [Value](#) property.

Inheritance Hierarchy

System.Object

[C1.C1Excel.XLShape](#)

C1.C1Excel.XLPictureShape

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Members](#)


[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)




The following tables list the members exposed by [XLPictureShape](#).














Public Constructors











	Name	Description
	XLPictureShape Constructor	Overloaded.

[Top](#)

Public Properties


	Name	Description
	BottomClip	Gets or sets the bottom clipping area of this XLPictureShape , in twips.
	Brightness	Gets or sets the brightness of this XLPictureShape (between 0 and 1).
	BrightnessInPercents	Gets or sets the brightness in percents of this XLPictureShape (between -

		100 and 100).
	Column	Gets the index of the column to which the shape is attached. (Inherited from C1.C1Excel.XLShape)
	Contrast	Gets or sets the contrast of this XLPictureShape (between 0 and 1).
	ContrastInPercents	Gets or sets the contrast in percents of this XLPictureShape (between -100 and 100).
	DashedLineStyle	Gets or sets the dash style of the line or border around the shape. (Inherited from C1.C1Excel.XLShape)
	FileName	Gets or sets the file name of this XLPictureShape .
	Hyperlink	Gets or sets the hyperlink associated with the shape. (Inherited from C1.C1Excel.XLShape)
	Id	Gets the unique identifier of the shape. (Inherited from C1.C1Excel.XLShape)
	Image	Gets a reference to the Image contained in this XLPictureShape .
	ImageSize	Gets image size in twips.
	IsEmpty	Overridden. Determines whether the shape is empty.
	LeftClip	Gets or sets the left clipping area of this XLPictureShape , in twips.
	LineColor	Gets or sets the color of the border around the shape. (Inherited from C1.C1Excel.XLShape)
	LineStyle	Gets or sets the style of the line or border around the shape. (Inherited from C1.C1Excel.XLShape)

	LineWidth	Gets or sets the width of the border around the shape, in twips. (Inherited from C1.C1Excel.XLShape)
	Rectangle	Gets or sets the rectangle that contains the shape, in twips. (Inherited from C1.C1Excel.XLShape)
	RightClip	Gets or sets the right clipping area of this XLPictureShape , in twips.
	Rotation	Gets or sets the rotation of the shape, in degrees. (Inherited from C1.C1Excel.XLShape)
	Row	Gets the index of the row to which the shape is attached. (Inherited from C1.C1Excel.XLShape)
	Sheet	Gets a reference to the parent XLSheet object. (Inherited from C1.C1Excel.XLShape)
	TopClip	Gets or sets the top clipping area of this XLPictureShape , in twips.
	Transparent	Gets or sets the transparent color of this XLPictureShape .
	ViewType	Gets or sets whether this XLPictureShape should be displayed in color, grayscale, or black and white.
	Workbook	Gets a reference to the parent C1XLBook object. (Inherited from C1.C1Excel.XLShape)

[Top](#)

Public Methods

	Name	Description
	Clone	Creates a new XLShape object that is a copy of the current instance. (Inherited from C1.C1Excel.XLShape)

[Top](#)

See Also

Reference

[XLPictureShape Class](#)

[C1.C1Excel Namespace](#)

XLPictureShape Constructor

Overload List

Overload	Description
XLPictureShape Constructor(Image,Int32,Int32)	Initializes a new instance of an XLPictureShape .
XLPictureShape Constructor(Image,Int32,Int32,Int32,Int32)	Initializes a new instance of an XLPictureShape .
XLPictureShape Constructor(Image,Rectangle)	Initializes a new instance of an XLPictureShape .
XLPictureShape Constructor(Image)	Initializes a new instance of an XLPictureShape .
XLPictureShape Constructor(Image,Size,ContentAlignment,ImageScaling)	Initializes a new instance of an XLPictureShape .
XLPictureShape Constructor(XLSheet,Image,Int32,Int32,Int32,Int32)	Initializes a new instance of an XLPictureShape .

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

[XLPictureShape Constructor \(Image, Int32, Int32\)](#)

[Example](#)

The image contained in the new [XLPictureShape](#).

The horizontal position of the new shape with respect to the cell, in twips.

The vertical position of the new shape with respect to the cell, in twips.

Initializes a new instance of an [XLPictureShape](#).

Syntax

Visual Basic (Declaration)	
<pre>Public Function New(_ ByVal img As System.Drawing.Image, _ ByVal x As System.Integer, _ ByVal y As System.Integer _)</pre>	
C#	
<pre>public XLPictureShape(System.Drawing.Image img, System.int x, System.int y)</pre>	

Parameters

img

The image contained in the new [XLPictureShape](#).

x

The horizontal position of the new shape with respect to the cell, in twips.

y

The vertical position of the new shape with respect to the cell, in twips.

Example

The code below adds an image to a cell. The image is rendered in its original size, and is indented from the top left corner of the cell by 30 twips:

- [C#](#)

```
// get sheet and cell
XLSheet sheet = c1ExcelBook1.Sheets[0];
XLCell cell = sheet[row, col];

// build XLPictureShape
XLPictureShape pic = new XLPictureShape(image, 30, 30);

// assign XLPictureShape to cell
cell.Value = pic;
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)
[XLPictureShape Members](#)
[Overload List](#)

XLPictureShape Constructor(Image,Int32,Int32,Int32,Int32)

[Example](#)

The image contained in the new [XLPictureShape](#).

The horizontal position of the image with respect to the cell, in twips.

The vertical position of the image with respect to the cell, in twips.

The width of the image, in twips.

The height of the image, in twips.

Initializes a new instance of an [XLPictureShape](#).

Syntax

Visual Basic (Declaration)

```
Public Function New( _  
    ByVal img As System.Drawing.Image, _  
    ByVal x As System.Integer, _  
    ByVal y As System.Integer, _  
    ByVal width As System.Integer, _  
    ByVal height As System.Integer _  
)
```

C#

```
public XLPictureShape(  
    System.Drawing.Image img,  
    System.int x,  
    System.int y,  
    System.int width,  
    System.int height  
)
```

Parameters

img

The image contained in the new [XLPictureShape](#).

x

The horizontal position of the image with respect to the cell, in twips.

y

The vertical position of the image with respect to the cell, in twips.

width

The width of the image, in twips.

height

The height of the image, in twips.

Example

The code below adds an image to a cell. The image is drawn within a rectangle centered on a cell with a 60 twip edge around it:

- [C#](#)

```
// get sheet and cell
XLSheet sheet = c1ExcelBook1.Sheets[0];
XLCell cell = sheet[row, col];

// calculate cell size to align picture
Rectangle rc = new Rectangle(0, 0
    sheet.Columns[col].Width,
    sheet.Rows[row].Height);

// add 60 twip edge
rc.Inflate(-60, -60);

// build XLPictureShape
XLPictureShape pic = new XLPictureShape(image,
    rc.X, rc.Y, rc.Width, rc.Height);

// assign XLPictureShape to cell
cell.Value = pic;
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)
[XLPictureShape Members](#)
[Overload List](#)

XLPictureShape Constructor(Image,Rectangle)

[Example](#)

The image contained in the new [XLPictureShape](#).

The rectangle that specifies the image size and position with respect to the cell, in twips.

Initializes a new instance of an [XLPictureShape](#).

Syntax

Visual Basic (Declaration)

```
Public Function New( _
    ByVal img As System.Drawing.Image, _
    ByVal rc As System.Drawing.Rectangle _
)
```

C#

```
public XLPictureShape(
    System.Drawing.Image img,
    System.Drawing.Rectangle rc
)
```

Parameters

img

The image contained in the new [XLPictureShape](#).

rc

The rectangle that specifies the image size and position with respect to the cell, in twips.

Example

The code below adds an image to a cell. The image is drawn within a rectangle centered on a cell with a 60 twip edge around it:

- [C#](#)

```
// get sheet and cell
XLSheet sheet = c1ExcelBook1.Sheets[0];
XLCell cell = sheet[row, col];

// calculate cell size to align picture
Rectangle rc = new Rectangle(0, 0,
    sheet.Columns[col].Width,
    sheet.Rows[row].Height);

// add 60 twip edge
rc.Inflate(-60, -60);

// build XLPictureShape
XLPictureShape pic = new XLPictureShape(image, rc);

// assign XLPictureShape to cell
cell.Value = pic;
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)
[XLPictureShape Members](#)
[Overload List](#)

XLPictureShape Constructor(Image)
The image contained in the new [XLPictureShape](#).
Initializes a new instance of an [XLPictureShape](#).

Syntax

Visual Basic (Declaration)	
<pre>Public Function New(_ ByVal img As System.Drawing.Image _)</pre>	
C#	
<pre>public XLPictureShape(System.Drawing.Image img)</pre>	

Parameters

img
The image contained in the new [XLPictureShape](#).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

[Overload List](#)

XLPictureShape Constructor(Image,Size,ContentAlignment,ImageScaling)

[Example](#)

The [Image](#) contained in the new [XLPictureShape](#).

The size of the cell that will contain the image, in pixels (used for aligning the image).

A **System.Drawing.ContentAlignment** value that specifies the position of the image in the cell.

An [ImageScaling](#) value that specifies the image scaling within the cell.

Initializes a new instance of an [XLPictureShape](#).

Syntax

Visual Basic (Declaration)

```
Public Function New( _  
    ByVal img As System.Drawing.Image, _  
    ByVal cellSize As System.Drawing.Size, _  
    ByVal align As System.Drawing.ContentAlignment, _  
    ByVal scale As ImageScaling _  
)
```

C#

```
public XLPictureShape(  
    System.Drawing.Image img,  
    System.Drawing.Size cellSize,  
    System.Drawing.ContentAlignment align,  
    ImageScaling scale  
)
```

Parameters

img

The [Image](#) contained in the new [XLPictureShape](#).

cellSize

The size of the cell that will contain the image, in pixels (used for aligning the image).

align

A **System.Drawing.ContentAlignment** value that specifies the position of the image in the cell.

scale

An [ImageScaling](#) value that specifies the image scaling within the cell.

Remarks

This constructor automatically calculates the image size, position, and clipping based on the cell and image sizes and on the given alignment and scaling parameters.

Example

The code below adds an image to a cell. The image is centered within the cell and scaled to fill the cell while preserving its aspect ratio.

- [C#](#)

```
// get sheet and cell
XLSheet sheet = c1ExcelBook1.Sheets[0];
XLCell cell = sheet[row, col];

// calculate cell size to align picture
Size cellSize = new Size(
    sheet.Columns[col].Width,
    sheet.Rows[row].Height);

// build XLPictureShape
XLPictureShape pic = new XLPictureShape(
    image,
    cellSize,
    ContentAlignment.MiddleCenter,
    ImageScaling.Scale);

// assign XLPictureShape to cell
cell.Value = pic;
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)
[XLPictureShape Members](#)
[Overload List](#)

XLPictureShape Constructor(XLSheet,Image,Int32,Int32,Int32,Int32)
[XLSheet](#) object that owns the new shape.

The image contained in the new [XLPictureShape](#).

The horizontal position of the image with respect to the sheet, in twips.

The vertical position of the image with respect to the sheet, in twips.

The width of the image, in twips.

The height of the image, in twips.

Initializes a new instance of an [XLPictureShape](#).

Syntax

Visual Basic (Declaration)

```
Public Function New( _  
    ByVal sheet As XLSheet, _  
    ByVal img As System.Drawing.Image, _  
    ByVal x As System.Integer, _  
    ByVal y As System.Integer, _  
    ByVal width As System.Integer, _  
    ByVal height As System.Integer _  
)
```

C#

```
public XLPictureShape(  
    XLSheet sheet,  
    System.Drawing.Image img,  
    System.int x,  
    System.int y,  
    System.int width,
```

```
System.int height  
)
```

Parameters

sheet

[XLSheet](#) object that owns the new shape.

img

The image contained in the new [XLPictureShape](#).

x

The horizontal position of the image with respect to the sheet, in twips.

y

The vertical position of the image with respect to the sheet, in twips.

width

The width of the image, in twips.

height

The height of the image, in twips.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2


















See Also










Reference

[XLPictureShape Class](#)
[XLPictureShape Members](#)
[Overload List](#)

Properties

>

Name	Description
 BottomClip	Gets or sets the bottom clipping area of this XLPictureShape , in twips.
 Brightness	Gets or sets the brightness of this XLPictureShape (between 0 and 1).
 BrightnessInPercents	Gets or sets the brightness in percents of this XLPictureShape (between -100 and 100).
 Column	Gets the index of the column to which the shape is attached. (Inherited from C1.C1Excel.XLSshape)
 Contrast	Gets or sets the contrast of this XLPictureShape (between 0 and 1).
 ContrastInPercents	Gets or sets the contrast in percents of this XLPictureShape (between -100 and 100).
 DashedLineStyle	Gets or sets the dash style of the line or border around the shape. (Inherited from C1.C1Excel.XLSshape)
 FileName	Gets or sets the file name of this XLPictureShape .
 Hyperlink	Gets or sets the hyperlink associated with the shape. (Inherited from C1.C1Excel.XLSshape)
 Id	Gets the unique identifier of the shape. (Inherited from C1.C1Excel.XLSshape)
 Image	Gets a reference to the Image contained in this XLPictureShape .
 ImageSize	Gets image size in twips.
 IsEmpty	Overridden. Determines whether the shape is empty.
 LeftClip	Gets or sets the left clipping area of this XLPictureShape , in twips.
 LineColor	Gets or sets the color of the border around the shape. (Inherited from C1.C1Excel.XLSshape)
 LineStyle	Gets or sets the style of the line or border around the shape. (Inherited from C1.C1Excel.XLSshape)
 LineWidth	Gets or sets the width of the border around the shape, in twips. (Inherited from C1.C1Excel.XLSshape)

 Rectangle	Gets or sets the rectangle that contains the shape, in twips. (Inherited from C1.C1Excel.XLShape)
 RightClip	Gets or sets the right clipping area of this XLPictureShape , in twips.
 Rotation	Gets or sets the rotation of the shape, in degrees. (Inherited from C1.C1Excel.XLShape)
 Row	Gets the index of the row to which the shape is attached. (Inherited from C1.C1Excel.XLShape)
 Sheet	Gets a reference to the parent XLSheet object. (Inherited from C1.C1Excel.XLShape)
 TopClip	Gets or sets the top clipping area of this XLPictureShape , in twips.
 Transparent	Gets or sets the transparent color of this XLPictureShape .
 ViewType	Gets or sets whether this XLPictureShape should be displayed in color, grayscale, or black and white.
 Workbook	Gets a reference to the parent C1XLBook object. (Inherited from C1.C1Excel.XLShape)

[Top](#)

See Also

Reference

[XLPictureShape Class](#)

[C1.C1Excel Namespace](#)

BottomClip Property

Gets or sets the bottom clipping area of this [XLPictureShape](#), in twips.

Syntax

Visual Basic (Declaration)	
Public Property BottomClip As System.Integer	
C#	

```
public System.Int BottomClip {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

Brightness Property

Gets or sets the brightness of this [XLPictureShape](#) (between 0 and 1).

Syntax

Visual Basic (Declaration)

```
Public Property Brightness As System.Single
```

C#

```
public System.Float Brightness {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

BrightnessInPercents Property

Gets or sets the brightness in percents of this [XLPictureShape](#) (between -100 and 100).

Syntax

Visual Basic (Declaration)	
Public Property BrightnessInPercents As System.Single	
C#	
public System.float BrightnessInPercents { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

Contrast Property

Gets or sets the contrast of this [XLPictureShape](#) (between 0 and 1).

Syntax

Visual Basic (Declaration)	
Public Property Contrast As System.Single	
C#	
public System.float Contrast { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

ContrastInPercents Property

Gets or sets the contrast in percents of this [XLPictureShape](#) (between -100 and 100).

Syntax

Visual Basic (Declaration)

```
Public Property ContrastInPercents As System.Single
```

C#

```
public System.float ContrastInPercents {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

FileName Property

Gets or sets the file name of this [XLPictureShape](#).

Syntax

Visual Basic (Declaration)

```
Public Property FileName As System.String
```

C#

```
public System.string FileName {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

Image Property

Gets a reference to the [Image](#) contained in this [XLPictureShape](#).

Syntax

Visual Basic (Declaration)	
Public ReadOnly Property Image As System.Drawing.Image	
C#	
public System.Drawing.Image Image { get ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

ImageSize Property

Gets image size in twips.

Syntax

Visual Basic (Declaration)	
----------------------------	--

Public ReadOnly Property ImageSize As System.Drawing.Size
C#
public System.Drawing.Size ImageSize {get;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

IsEmpty Property

Determines whether the shape is empty.

Syntax

Visual Basic (Declaration)
Public Overrides ReadOnly Property IsEmpty As System.Boolean
C#
public override System.bool IsEmpty {get;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

LeftClip Property

Gets or sets the left clipping area of this [XLPictureShape](#), in twips.

Syntax

Visual Basic (Declaration)

```
Public Property LeftClip As System.Integer
```

C#

```
public System.int LeftClip {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

RightClip Property

Gets or sets the right clipping area of this [XLPictureShape](#), in twips.

Syntax

Visual Basic (Declaration)

```
Public Property RightClip As System.Integer
```

C#

```
public System.int RightClip {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

TopClip Property

Gets or sets the top clipping area of this [XLPictureShape](#), in twips.

Syntax

Visual Basic (Declaration)	
Public Property TopClip As System.Integer	
C#	
public System.int TopClip { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

Transparent Property

Gets or sets the transparent color of this [XLPictureShape](#).

Syntax

Visual Basic (Declaration)	
Public Property Transparent As System.Drawing.Color	
C#	

```
public System.Drawing.Color Transparent {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

ViewType Property

Gets or sets whether this [XLPictureShape](#) should be displayed in color, grayscale, or black and white.

Syntax

Visual Basic (Declaration)	
<pre>Public Property ViewType As XLPictureViewType</pre>	
C#	
<pre>public XLPictureViewType ViewType {get; set;}</pre>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPictureShape Class](#)

[XLPictureShape Members](#)

XLPrintSettings

Provides options and settings for printing [XLSheet](#) objects.

Object Model

XLPrintSettings

Syntax

Visual Basic (Declaration)	
Public Class XLPrintSettings	
C#	
public class XLPrintSettings	

Remarks

The settings are applied to each sheet and are accessible through the sheet's [XLSheet.PrintSettings](#) property.

Note that [C1XLBook](#) does not provide any printing services. The settings are used when printing the sheet from Excel.

Inheritance Hierarchy

System.Object

C1.C1Excel.XLPrintSettings

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Members](#)

[C1.C1Excel Namespace](#)

Overview

Provides options and settings for printing [XLSheet](#) objects.

Object Model

XLPrintSettings

Syntax

Visual Basic (Declaration)	
Public Class XLPrintSettings	
C#	
public class XLPrintSettings	

Remarks

The settings are applied to each sheet and are accessible through the sheet's [XLSheet.PrintSettings](#) property.

Note that [C1XLBook](#) does not provide any printing services. The settings are used when printing the sheet from Excel.

Inheritance Hierarchy

System.Object

C1.C1Excel.XLPrintSettings

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Members](#)


[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)











The following tables list the members exposed by [XLPrintSettings](#).













Public Constructors








	Name	Description
	XLPrintSettings Constructor	Creates a new instance of the XLPrintSettings class.

[Top](#)

Public Properties


	Name	Description
	AutoScale	Gets or sets the scaling mode used for printed output.
	BlackAndWhite	Gets or sets whether to print the sheet in monochrome mode.
	CenterHorizontal	Gets or sets whether the sheet should be centered horizontally on the page when printed.
	CenterVertical	Gets or sets whether the sheet should be centered vertically on the page when printed.
	Copies	Gets or sets the number of copies to print.
	DraftQuality	Gets or sets whether to print the sheet in draft quality mode.
	FitPagesAcross	Fit the sheet to this number of pages across (0 means use as many as needed).
	FitPagesDown	Fit the sheet to this number of pages down (0 means use as many as needed).
	Footer	Gets or sets the string to be displayed as a page footer when the sheet is printed.
	FooterPictureCenter	Gets or sets the System.Drawing.Image or XLPictureShape for the

		center part of the footer.
	FooterPictureLeft	Gets or sets the System.Drawing.Image or XLPictureShape for the left part of the footer.
	FooterPictureRight	Gets or sets the System.Drawing.Image or XLPictureShape for the right part of the footer.
	Header	Gets or sets the string to be displayed as a page header when the sheet is printed.
	HeaderPictureCenter	Gets or sets the System.Drawing.Image or XLPictureShape for the center part of the header.
	HeaderPictureLeft	Gets or sets the System.Drawing.Image or XLPictureShape for the left part of the header.
	HeaderPictureRight	Gets or sets the System.Drawing.Image or XLPictureShape for the right part of the header.
	Landscape	Gets or sets whether to print the sheet in landscape mode.
	MarginBottom	Gets or sets the bottom margin, in inches. Set to a negative value to use the default margin.
	MarginFooter	Gets or sets the footer margin, in inches. Set to a negative value to use the default margin.
	MarginHeader	Gets or sets the header margin, in inches. Set to a negative value to use the default margin.
	MarginLeft	Gets or sets the left margin, in inches. Set to a negative value to use the default margin.
	MarginRight	Gets or sets the right margin, in inches. Set to a negative value to use the

		default margin.
	MarginTop	Gets or sets the top margin, in inches. Set to a negative value to use the default margin.
	PaperKind	Gets or sets the paper size to use when printing the sheet.
	PrintGridlines	Gets or sets whether the gridlines will be printed.
	PrintHeaders	Gets or sets whether row and column headers (the areas with row numbers and column letters) will be printed.
	PrintPagesInRows	Gets or sets whether to print the pages in rows (across first) or in columns (down first).
	ScalingFactor	Gets or sets the scaling factor (in percent) to use when printing the sheet.
	StartPage	Gets or sets the initial page number to use when printing the sheet.

[Top](#)

Public Methods

	Name	Description
	Clone	Creates a new XLPrintSettings object that is a copy of the current instance.

[Top](#)

See Also

Reference

[XLPrintSettings Class](#)

[C1.C1Excel Namespace](#)

XLPrintSettings Constructor

Creates a new instance of the [XLPrintSettings](#) class.

Syntax

Visual Basic (Declaration)	
Public Function New()	
C#	
public XLPrintSettings()	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference


[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

Methods

For a list of all members of this type, see [XLPrintSettings members](#).

Public Methods

	Name	Description
	Clone	Creates a new XLPrintSettings object that is a copy of the current instance.

[Top](#)

See Also

Reference

[XLPrintSettings Class](#)

[C1.C1Excel Namespace](#)

Clone Method

Creates a new [XLPrintSettings](#) object that is a copy of the current instance.

Syntax

Visual Basic (Declaration)	
Public Function Clone() As XLPrintSettings	
C#	
public XLPrintSettings Clone()	

Return Value

A new [XLPrintSettings](#) object that is a copy of the current instance.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also






Reference
















[XLPrintSettings Class](#)










[XLPrintSettings Members](#)

Properties

>

Name	Description
 AutoScale	Gets or sets the scaling mode used for printed output.
 BlackAndWhite	Gets or sets whether to print the sheet in monochrome mode.
 CenterHorizontal	Gets or sets whether the sheet should be centered horizontally on the page when printed.
 CenterVertical	Gets or sets whether the sheet should be centered vertically on the page when printed.
 Copies	Gets or sets the number of copies to print.

 DraftQuality	Gets or sets whether to print the sheet in draft quality mode.
 FitPagesAcross	Fit the sheet to this number of pages across (0 means use as many as needed).
 FitPagesDown	Fit the sheet to this number of pages down (0 means use as many as needed).
 Footer	Gets or sets the string to be displayed as a page footer when the sheet is printed.
 FooterPictureCenter	Gets or sets the System.Drawing.Image or XLPictureShape for the center part of the footer.
 FooterPictureLeft	Gets or sets the System.Drawing.Image or XLPictureShape for the left part of the footer.
 FooterPictureRight	Gets or sets the System.Drawing.Image or XLPictureShape for the right part of the footer.
 Header	Gets or sets the string to be displayed as a page header when the sheet is printed.
 HeaderPictureCenter	Gets or sets the System.Drawing.Image or XLPictureShape for the center part of the header.
 HeaderPictureLeft	Gets or sets the System.Drawing.Image or XLPictureShape for the left part of the header.
 HeaderPictureRight	Gets or sets the System.Drawing.Image or XLPictureShape for the right part of the header.
 Landscape	Gets or sets whether to print the sheet in landscape mode.
 MarginBottom	Gets or sets the bottom margin, in inches. Set to a negative value to use the default margin.
 MarginFooter	Gets or sets the footer margin, in inches. Set to a negative value to use the default margin.
 MarginHeader	Gets or sets the header margin, in inches. Set to a negative value to use the default margin.

 MarginLeft	Gets or sets the left margin, in inches. Set to a negative value to use the default margin.
 MarginRight	Gets or sets the right margin, in inches. Set to a negative value to use the default margin.
 MarginTop	Gets or sets the top margin, in inches. Set to a negative value to use the default margin.
 PaperKind	Gets or sets the paper size to use when printing the sheet.
 PrintGridlines	Gets or sets whether the gridlines will be printed.
 PrintHeaders	Gets or sets whether row and column headers (the areas with row numbers and column letters) will be printed.
 PrintPagesInRows	Gets or sets whether to print the pages in rows (across first) or in columns (down first).
 ScalingFactor	Gets or sets the scaling factor (in percent) to use when printing the sheet.
 StartPage	Gets or sets the initial page number to use when printing the sheet.

[Top](#)

See Also

Reference

[XLPrintSettings Class](#)

[C1.C1Excel Namespace](#)

AutoScale Property

Gets or sets the scaling mode used for printed output.

Syntax

Visual Basic (Declaration)	
Public Property AutoScale As System.Boolean	
C#	
public System.bool AutoScale { get ; set ;}	

Remarks

If [AutoScale](#) is set to true, then the printed sheet will be automatically scaled to fit the number of pages specified by the [FitPagesAcross](#) and [FitPagesDown](#) properties.

If [AutoScale](#) is set to false, then the printed sheet will be scaled according to the value of the [ScalingFactor](#) property.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

BlackAndWhite Property

Gets or sets whether to print the sheet in monochrome mode.

Syntax

Visual Basic (Declaration)	
Public Property BlackAndWhite As System.Boolean	
C#	
public System.bool BlackAndWhite { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

CenterHorizontal Property

Gets or sets whether the sheet should be centered horizontally on the page when printed.

Syntax

Visual Basic (Declaration)	
Public Property CenterHorizontal As System.Boolean	
C#	
public System.bool CenterHorizontal { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

CenterVertical Property

Gets or sets whether the sheet should be centered vertically on the page when printed.

Syntax

Visual Basic (Declaration)	
Public Property CenterVertical As System.Boolean	
C#	
public System.bool CenterVertical { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

Copies Property

Gets or sets the number of copies to print.

Syntax

Visual Basic (Declaration)	
Public Property Copies As System.Integer	
C#	
public System.int Copies { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

DraftQuality Property

Gets or sets whether to print the sheet in draft quality mode.

Syntax

Visual Basic (Declaration)	
----------------------------	--

Public Property DraftQuality As System.Boolean
C#
public System.bool DraftQuality {get; set;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

FitPagesAcross Property

Fit the sheet to this number of pages across (0 means use as many as needed).

Syntax

Visual Basic (Declaration)
Public Property FitPagesAcross As System.Integer
C#
public System.int FitPagesAcross {get; set;}

Remarks

Causes C1Excel to select "fit to page" print mode, ignoring the value of the [ScalingFactor](#) property. Setting the [FitPagesAcross](#) or [FitPagesDown](#) properties automatically sets the [AutoScale](#) property to true, causing C1Excel to calculate the scaling factor based on the given number of pages and to ignore the value of the [ScalingFactor](#) property.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

FitPagesDown Property

Fit the sheet to this number of pages down (0 means use as many as needed).

Syntax

Visual Basic (Declaration)	
Public Property FitPagesDown As System.Integer	
C#	
public System.int FitPagesDown { get ; set ;}	

Remarks

Setting the [FitPagesAcross](#) or [FitPagesDown](#) properties automatically sets the [AutoScale](#) property to true, causing C1Excel to calculate the scaling factor based on the given number of pages and to ignore the value of the [ScalingFactor](#) property.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

Footer Property

Gets or sets the string to be displayed as a page footer when the sheet is printed.

Syntax

Visual Basic (Declaration)	
Public Property Footer As System.String	
C#	
public System. string Footer { get ; set ;}	

Remarks

The footer string has the same structure and embedded commands as the header string. See the [Header](#) property for details.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

FooterPictureCenter Property

Gets or sets the **System.Drawing.Image** or [XLPictureShape](#) for the center part of the footer.

Syntax

Visual Basic (Declaration)	
Public Property FooterPictureCenter As System.Object	
C#	
public System. object FooterPictureCenter { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

FooterPictureLeft Property

Gets or sets the **System.Drawing.Image** or [XLPictureShape](#) for the left part of the footer.

Syntax

Visual Basic (Declaration)	
Public Property FooterPictureLeft As System.Object	
C#	
public System.object FooterPictureLeft { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

FooterPictureRight Property

Gets or sets the **System.Drawing.Image** or [XLPictureShape](#) for the right part of the footer.

Syntax

Visual Basic (Declaration)	
Public Property FooterPictureRight As System.Object	
C#	

```
public System.Object FooterPictureRight {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

Header Property

[Example](#)

Gets or sets the string to be displayed as a page header when the sheet is printed.

Syntax

Visual Basic (Declaration)

```
Public Property Header As System.String
```

C#

```
public System.string Header {get; set;}
```

Remarks

The header string may contain special commands, i.e. placeholders for the page number, current date, or text formatting attributes. Most of these fields are represented by single letters with a leading ampersand ("&").

The page header is divided into 3 sections: left, center, and right. Each section is introduced by a special command ("&L", "&C", and "&R"). All text and all commands following are part of the selected section.

The following commands are available:

&L Start of the left section

&C Start of the centered section

&R Start of the right section

&P Current page number

&N Page count

&D Current date

&T Current time

&A Sheet name

&F File name without path

&Z File path without file name

&G Picture (file name)

&B Bold toggle

&I Italic toggle

&U Underline toggle

&E Double underline toggle

&S Strikeout toggle

&X Superscript toggle

&Y Subscript toggle

&"[FontName]" Set new font

&"[FontName,FontStyle]" Set new font with specified style. The style is in most cases "Regular", "Bold", "Italic", or "Bold Italic".

&[fontheight] Set font height in points.

Example

The code below creates a header with left, center, and right portions.

- C#

```
PrintSettings ps = sheet.PrintSettings;  
ps.Header = "&LHeader Left&CHheader Center&RHeader Right";
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

HeaderPictureCenter Property

Gets or sets the **System.Drawing.Image** or [XLPictureShape](#) for the center part of the header.

Syntax

Visual Basic (Declaration)	
Public Property HeaderPictureCenter As System.Object	
C#	
public System.object HeaderPictureCenter { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

HeaderPictureLeft Property

Gets or sets the **System.Drawing.Image** or [XLPictureShape](#) for the left part of the header.

Syntax

Visual Basic (Declaration)	
----------------------------	--

Public Property HeaderPictureLeft As System.Object
C#
public System.object HeaderPictureLeft {get; set;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

HeaderPictureRight Property

Gets or sets the **System.Drawing.Image** or [XLPictureShape](#) for the right part of the header.

Syntax

Visual Basic (Declaration)
Public Property HeaderPictureRight As System.Object
C#
public System.object HeaderPictureRight {get; set;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

Landscape Property

Gets or sets whether to print the sheet in landscape mode.

Syntax

Visual Basic (Declaration)

```
Public Property Landscape As System.Boolean
```

C#

```
public System.bool Landscape {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

MarginBottom Property

Gets or sets the bottom margin, in inches. Set to a negative value to use the default margin.

Syntax

Visual Basic (Declaration)

```
Public Property MarginBottom As System.Double
```

C#

```
public System.double MarginBottom {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

MarginFooter Property

Gets or sets the footer margin, in inches. Set to a negative value to use the default margin.

Syntax

Visual Basic (Declaration)	
Public Property MarginFooter As System.Double	
C#	
public System.double MarginFooter { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

MarginHeader Property

Gets or sets the header margin, in inches. Set to a negative value to use the default margin.

Syntax

Visual Basic (Declaration)	
Public Property MarginHeader As System.Double	
C#	

```
public System.double MarginHeader {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

MarginLeft Property

Gets or sets the left margin, in inches. Set to a negative value to use the default margin.

Syntax

Visual Basic (Declaration)

```
Public Property MarginLeft As System.Double
```

C#

```
public System.double MarginLeft {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

MarginRight Property

Gets or sets the right margin, in inches. Set to a negative value to use the default margin.

Syntax

Visual Basic (Declaration)	
<code>Public Property MarginRight As System.Double</code>	
C#	
<code>public System.double MarginRight {get; set;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

MarginTop Property

Gets or sets the top margin, in inches. Set to a negative value to use the default margin.

Syntax

Visual Basic (Declaration)	
<code>Public Property MarginTop As System.Double</code>	
C#	
<code>public System.double MarginTop {get; set;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

PaperKind Property

Gets or sets the paper size to use when printing the sheet.

Syntax

Visual Basic (Declaration)	
Public Property PaperKind As System.Drawing.Printing.PaperKind	
C#	
public System.Drawing.Printing.PaperKind PaperKind { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

PrintGridlines Property

Gets or sets whether the gridlines will be printed.

Syntax

Visual Basic (Declaration)	
Public Property PrintGridlines As System.Boolean	
C#	
public System. bool PrintGridlines { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

PrintHeaders Property

Gets or sets whether row and column headers (the areas with row numbers and column letters) will be printed.

Syntax

Visual Basic (Declaration)

```
Public Property PrintHeaders As System.Boolean
```

C#

```
public System.bool PrintHeaders {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

PrintPagesInRows Property

Gets or sets whether to print the pages in rows (across first) or in columns (down first).

Syntax

Visual Basic (Declaration)	
Public Property PrintPagesInRows As System.Boolean	
C#	
public System.bool PrintPagesInRows { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

ScalingFactor Property

Gets or sets the scaling factor (in percent) to use when printing the sheet.

Syntax

Visual Basic (Declaration)	
Public Property ScalingFactor As System.Integer	
C#	
public System.int ScalingFactor { get ; set ;}	

Remarks

Setting the [ScalingFactor](#) property automatically sets the [AutoScale](#) property to false, causing C1Excel to use the selected scaling factor and to ignore the value of the [FitPagesAcross](#) and [FitPagesDown](#) properties.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

StartPage Property

Gets or sets the initial page number to use when printing the sheet.

Syntax

Visual Basic (Declaration)	
Public Property StartPage As System.Integer	
C#	
public System.int StartPage { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLPrintSettings Class](#)

[XLPrintSettings Members](#)

XLRow

Represents a row in a worksheet, provides properties for setting the row's height, style, and visibility.

Object Model

XLRow

Syntax

Visual Basic (Declaration)	
<code>Public Class XRow</code>	
C#	
<code>public class XRow</code>	

Inheritance Hierarchy

System.Object

C1.C1Excel.XRow

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XRow Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents a row in a worksheet, provides properties for setting the row's height, style, and visibility.

Object Model

XRow

Syntax

Visual Basic (Declaration)	
<code>Public Class XRow</code>	

C#

```
public class XRow
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XRow

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XRow Members](#)


[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)



The following tables list the members exposed by [XRow](#).








Public Constructors

	Name	Description
	XRow Constructor	Creates a new instance of the XRow class.

[Top](#)


Public Properties

	Name	Description
	Book	Gets a reference to the parent C1XLBook object.
	Collapsed	Gets or sets collapsed flag for the row.

	Height	Gets or sets the height of the row, in twips.
	IsSubtotal	Gets whether the row is subtotal.
	OutlineLevel	Gets or sets subtotal outline level for the row.
	PageBreak	Gets or sets whether there will be a forced page break after this row.
	Sheet	Gets a reference to the parent XLSheet object.
	Style	Gets or sets the XLStyle object that determines the appearance of the row.
	Visible	Gets or sets whether the row is visible.

[Top](#)

Public Methods

	Name	Description
	Clone	Creates a new XLRow object that is a copy of the current instance.

[Top](#)

See Also

Reference

[XLRow Class](#)

[C1.C1Excel Namespace](#)

XLRow Constructor

Creates a new instance of the [XLRow](#) class.

Syntax

Visual Basic (Declaration)

```
Public Function New()
```

C#

```
public XLRow()
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference


[XLRow Class](#)

[XLRow Members](#)

Methods

For a list of all members of this type, see [XLRow members](#).

Public Methods

	Name	Description
	Clone	Creates a new XLRow object that is a copy of the current instance.

[Top](#)

See Also

Reference

[XLRow Class](#)

[C1.C1Excel Namespace](#)

Clone Method

Creates a new [XLRow](#) object that is a copy of the current instance.

Syntax

Visual Basic (Declaration)

```
Public Function Clone() As XLRow
```

C#

```
public XlRow Clone()
```

Return Value

A new [XlRow](#) object that is a copy of the current instance.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference







[XlRow Class](#)




[XlRow Members](#)

Properties

For a list of all members of this type, see [XlRow members](#).

Public Properties

	Name	Description
	Book	Gets a reference to the parent C1XlBook object.
	Collapsed	Gets or sets collapsed flag for the row.
	Height	Gets or sets the height of the row, in twips.
	IsSubtotal	Gets whether the row is subtotal.
	OutlineLevel	Gets or sets subtotal outline level for the row.
	PageBreak	Gets or sets whether there will be a forced page break after this row.

	Sheet	Gets a reference to the parent XLSheet object.
	Style	Gets or sets the XLStyle object that determines the appearance of the row.
	Visible	Gets or sets whether the row is visible.

[Top](#)

See Also

Reference

[XLRow Class](#)

[C1.C1Excel Namespace](#)

Book Property

Gets a reference to the parent [C1XLBook](#) object.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Book As C1XLBook</code>	
C#	
<code>public C1XLBook Book {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRow Class](#)

[XLRow Members](#)

Collapsed Property

Gets or sets collapsed flag for the row.

Syntax

Visual Basic (Declaration)	
Public Property Collapsed As System.Boolean	
C#	
public System.bool Collapsed { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRow Class](#)

[XLRow Members](#)

Height Property

Gets or sets the height of the row, in twips.

Syntax

Visual Basic (Declaration)	
Public Property Height As System.Integer	
C#	
public System.int Height { get ; set ;}	

Remarks

A value of -1 causes the row to be displayed using the sheet's [XLSheet.DefaultRowHeight](#) if cells of the row not contain data, otherwise this height of the row depended from used font of the data.

To convert between pixels and twips, use the [C1XLBook.TwipsToPixels](#) and [PixelsToTwips](#) methods.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRow Class](#)

[XLRow Members](#)

IsSubtotal Property

Gets whether the row is subtotal.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property IsSubtotal As System.Boolean</code>	
C#	
<code>public System.bool IsSubtotal {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRow Class](#)

[XLRow Members](#)

OutlineLevel Property

Gets or sets subtotal outline level for the row.

Syntax

Visual Basic (Declaration)	
----------------------------	--

Public Property OutlineLevel As System.Integer
C#
public System.int OutlineLevel {get; set;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRow Class](#)

[XLRow Members](#)

PageBreak Property

[Example](#)

Gets or sets whether there will be a forced page break after this row.

Syntax

Visual Basic (Declaration)
Public Property PageBreak As System.Boolean
C#
public System.bool PageBreak {get; set;}

Example

The code below inserts forced page breaks at every 10th row on a sheet, and clears the breaks at all other rows.

- [C#](#)

```
C1.C1Excel.XLSheet sheet = c1XLBook1.Sheets[0];
for (int r = 0; r < sheet.Rows.Count; r++)
{
    sheet.Rows[r].PageBreak = (r > 0 && r % 10 == 0);
}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRow Class](#)

[XLRow Members](#)

Sheet Property

Gets a reference to the parent [XLSheet](#) object.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Sheet As XLSheet</code>	
C#	
<code>public XLSheet Sheet {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRow Class](#)

[XLRow Members](#)

Style Property

Gets or sets the [XLStyle](#) object that determines the appearance of the row.

Syntax

Visual Basic (Declaration)	
Public Property Style As XLStyle	
C#	
public XLStyle Style { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRow Class](#)

[XLRow Members](#)

Visible Property

Gets or sets whether the row is visible.

Syntax

Visual Basic (Declaration)	
Public Property Visible As System.Boolean	
C#	
public System.bool Visible { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRow Class](#)
[XLRow Members](#)

XLRowCollection
[Example](#)

Represents a collection of [XLRow](#) objects that represent the individual rows in each [XLSheet](#).

Object Model

XLRowCollection

Syntax

Visual Basic (Declaration)	
<pre><System.Reflection.DefaultMemberAttribute("Item")> Public Class XLRowCollection</pre>	
C#	
<pre>[System.Reflection.DefaultMemberAttribute("Item")] public class XLRowCollection</pre>	

Remarks

The collection has methods for counting, enumerating, adding, and removing rows from the collection.

Example

Note that you can create rows automatically by using the sheet's indexer. For example, the following code retrieves the cell at coordinates (3,3) and in doing so automatically creates four rows and four columns automatically:

- [C#](#)

```
C1XLBook book = new C1XLBook();
sheet = book.Sheets[0];
XLCell cell = sheet[3,3]; // creates 4 rows and 4 columns
```

Inheritance Hierarchy

System.Object
C1.C1Excel.XLRowCollection

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRowCollection Members](#)

[C1.C1Excel Namespace](#)

[Overview](#)

[Example](#)

Represents a collection of [XLRow](#) objects that represent the individual rows in each [XLSheet](#).

Object Model

XLRowCollection

Syntax

Visual Basic (Declaration)	
<pre><System.Reflection.DefaultMemberAttribute("Item")> Public Class XLRowCollection</pre>	
C#	
<pre>[System.Reflection.DefaultMemberAttribute("Item")] public class XLRowCollection</pre>	

Remarks

The collection has methods for counting, enumerating, adding, and removing rows from the collection.

Example

Note that you can create rows automatically by using the sheet's indexer. For example, the following code retrieves the cell at coordinates (3,3) and in doing so automatically creates four rows and four columns automatically:

- [C#](#)

```
C1XLBook book = new C1XLBook();
sheet = book.Sheets[0];
XLCe11 cell = sheet[3,3]; // creates 4 rows and 4 columns
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLRowCollection

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRowCollection Members](#)






[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)

The following tables list the members exposed by [XLRowCollection](#).

Public Properties

	Name	Description
	Book	Gets a reference to the parent C1XLBook object.
	Count	Gets the number of items in the collection.
	Frozen	Gets or sets the number of frozen rows in the collection.
	Item	Returns a reference to the XLRow object at the specified index.
	Sheet	Gets a reference to the parent XLSheet object.

[Top](#)

Public Methods

	Name	Description
≡	Add	Overloaded. Creates a new XLRow object and adds it to the collection.
≡	Clear	Removes all items from the collection.
≡	Contains	Determines whether an XLRow is a member of the collection.
≡	IndexOf	Gets the index of a given XLRow object in the collection.
≡	Insert	Overloaded. Creates a new XLRow object and inserts it at a specific position in the collection.
≡	Remove	Removes an XLRow object from the collection.
≡	RemoveAt	Removes the XLRow object at a given position from the collection.

[Top](#)

See Also

Reference

[XLRowCollection Class](#)






[C1.C1Excel Namespace](#)

Methods

For a list of all members of this type, see [XLRowCollection members](#).

Public Methods

	Name	Description
≡	Add	Overloaded. Creates a new XLRow object and adds it to the collection.
≡	Clear	Removes all items from the collection.

≡  Contains	Determines whether an XLRow is a member of the collection.
≡  IndexOf	Gets the index of a given XLRow object in the collection.
≡  Insert	Overloaded. Creates a new XLRow object and inserts it at a specific position in the collection.
≡  Remove	Removes an XLRow object from the collection.
≡  RemoveAt	Removes the XLRow object at a given position from the collection.

[Top](#)

See Also

Reference

[XLRowCollection Class](#)

[C1.C1Excel Namespace](#)

Add Method

Creates a new [XLRow](#) object and adds it to the collection.

Overload List

Overload	Description
Add()	Creates a new XLRow object and adds it to the collection.
Add(XLRow)	Adds an XLRow object to the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRowCollection Class](#)

[XLRowCollection Members](#)

Add() Method

Creates a new [XLRow](#) object and adds it to the collection.

Syntax

Visual Basic (Declaration)	
<code>Public Overloads Function Add() As XLRow</code>	
C#	
<code>public XLRow Add()</code>	

Return Value

A reference to the new [XLRow](#) object.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRowCollection Class](#)

[XLRowCollection Members](#)

[Overload List](#)

Add(XLRow) Method

The item to add to the collection.

Adds an [XLRow](#) object to the collection.

Syntax

Visual Basic (Declaration)	
<code>Public Overloads Function Add(_</code>	

<pre> ByVal row As XRow _) As XRow </pre>	
C#	
<pre> public XRow Add(XRow row) </pre>	

Parameters

row

The item to add to the collection.

Return Value

A reference to the item that was added to the collection (in this case, always the [row](#) parameter).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XRowCollection Class](#)
[XRowCollection Members](#)
[Overload List](#)

Clear Method

Removes all items from the collection.

Syntax

Visual Basic (Declaration)	
<pre> Public Sub Clear() </pre>	
C#	

```
public void Clear()
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRowCollection Class](#)

[XLRowCollection Members](#)

Contains Method

Item to look for.

Determines whether an [XLRow](#) is a member of the collection.

Syntax

Visual Basic (Declaration)

```
Public Function Contains( _  
    ByVal row As XLRow _  
) As System.Boolean
```

C#

```
public System.bool Contains(  
    XLRow row  
)
```

Parameters

row

Item to look for.

Return Value

True if the collection contains the item, False otherwise.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRowCollection Class](#)
[XLRowCollection Members](#)

IndexOf Method

Item to look for.

Gets the index of a given [XLRow](#) object in the collection.

Syntax

Visual Basic (Declaration)

```
Public Function IndexOf( _  
    ByVal row As XLRow _  
) As System.Integer
```

C#

```
public System.int IndexOf(  
    XLRow row  
)
```

Parameters

row

Item to look for.

Return Value

The position of the item in the collection, or -1 if the item is not a member of the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRowCollection Class](#)

[XLRowCollection Members](#)

Insert Method

Creates a new [XLRow](#) object and inserts it at a specific position in the collection.

Overload List

Overload	Description
Insert(Int32)	Creates a new XLRow object and inserts it at a specific position in the collection.
Insert(Int32,XLRow)	Inserts an XLRow object at a specific position in the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRowCollection Class](#)

[XLRowCollection Members](#)

Insert(Int32) Method

Position where the new item will be inserted.

Creates a new [XLRow](#) object and inserts it at a specific position in the collection.

Syntax

Visual Basic (Declaration)	
Public Overloads Function Insert(_	

```
ByVal index As System.Integer _  
) As XLRRow
```

C#

```
public XLRRow Insert(  
    System.int index  
)
```

Parameters

index

Position where the new item will be inserted.

Return Value

A reference to the new item.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRRowCollection Class](#)
[XLRRowCollection Members](#)
[Overload List](#)

Insert(Int32,XLRRow) Method

Position where the item will be inserted.

Item that will be inserted.

Inserts an [XLRRow](#) object at a specific position in the collection.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Insert( _
```



```
    ByVal index As System.Integer, _  
    ByVal row As XLRW _  
) As XLRW
```

C#

```
public XLRW Insert(  
    System.int index,  
    XLRW row  
)
```

Parameters

index

Position where the item will be inserted.

row

Item that will be inserted.

Return Value

A reference to the item that was added to the collection.

Remarks

The maximum number of [XLRW](#) objects in a [XLSheet](#) is 65,536. This is a limitation imposed by Excel 2003 and below.

For Excel 2007 and above, the maximum number of [XLRW](#) objects in an [XLSheet](#) is 1,048,576.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRWCollection Class](#)

[XLRWCollection Members](#)

[Overload List](#)

Remove Method

Item to be removed from the collection.

Removes an [XLRow](#) object from the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Function Remove(_ ByVal row As XLRow _) As XLRow</pre>	
C#	
<pre>public XLRow Remove(XLRow row)</pre>	

Parameters

row

Item to be removed from the collection.

Return Value

A reference to the item that was removed.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRowCollection Class](#)

[XLRowCollection Members](#)

RemoveAt Method

Index of the item to remove from the collection.

Removes the [XLRow](#) object at a given position from the collection.

Syntax

Visual Basic (Declaration)

```
Public Function RemoveAt( _  
    ByVal index As System.Integer _  
) As XLRRow
```

C#

```
public XLRRow RemoveAt(  
    System.int index  
)
```

Parameters

index

Index of the item to remove from the collection.

Return Value

A reference to the item that was removed from the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRRowCollection Class](#)






[XLRRowCollection Members](#)

Properties

For a list of all members of this type, see [XLRRowCollection members](#).

Public Properties

Name	Description
------	-------------

	Book	Gets a reference to the parent C1XLBook object.
	Count	Gets the number of items in the collection.
	Frozen	Gets or sets the number of frozen rows in the collection.
	Item	Returns a reference to the XLRow object at the specified index.
	Sheet	Gets a reference to the parent XLSheet object.

[Top](#)

See Also

Reference

[XLRowCollection Class](#)

[C1.C1Excel Namespace](#)

Book Property

Gets a reference to the parent [C1XLBook](#) object.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Book As C1XLBook</code>	
C#	
<code>public C1XLBook Book {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRowCollection Class](#)

[XLRowCollection Members](#)

Count Property

Gets the number of items in the collection.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property Count As System.Integer
```

C#

```
public System.int Count {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRowCollection Class](#)

[XLRowCollection Members](#)

Frozen Property

Gets or sets the number of frozen rows in the collection.

Syntax

Visual Basic (Declaration)

```
Public Property Frozen As System.Integer
```

C#

```
public System.int Frozen {get; set;}
```

Remarks

Frozen rows are displayed on the top of the sheet and do not scroll vertically. They are useful for displaying column headers.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRowCollection Class](#)

[XLRowCollection Members](#)

Item Property

Returns a reference to the [XLRow](#) object at the specified index.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Default Property Item( _  
    ByVal index As System.Integer _  
) As XLRow
```

C#

```
public XLRow this[  
    System.int index  
]; {get;}
```

Parameters

index

Remarks

The indexer will create a new [XLRow](#) object at the specified position if necessary. It never returns null.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRowCollection Class](#)
[XLRowCollection Members](#)

Sheet Property

Gets a reference to the parent [XLSheet](#) object.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Sheet As XLSheet</code>	
C#	
<code>public XLSheet Sheet {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLRowCollection Class](#)
[XLRowCollection Members](#)

XLShape

Represents a shape embedded in an [XLSheet](#).

Object Model

XLShape

Syntax

Visual Basic (Declaration)	
Public MustInherit Class XLShape	
C#	
public abstract class XLShape	

Remarks

Excel sheets may contain many types of embedded shapes, including images, graphical elements, controls, and comments.

The **XLShape** abstract class contains information that is common to all shape types, including the shape's location ([Sheet](#), [Row](#), [Column](#), [Rectangle](#), [Rotation](#)), the type of border drawn around the shape ([LineWidth](#), [LineColor](#), [LineStyle](#)), the [Rotation](#) applied to the shape, and [Hyperlink](#) information.

The [XLPictureShape](#) class derives from **XLShape** and is used to embed images in sheets.

Inheritance Hierarchy

System.Object

C1.C1Excel.XLShape

[C1.C1Excel.XLOpaqueShape](#)

[C1.C1Excel.XLPictureShape](#)

[C1.C1Excel.XLTextShape](#)

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents a shape embedded in an [XLSheet](#).

Object Model

XLShape

Syntax

Visual Basic (Declaration)	
Public MustInherit Class XLShape	
C#	
public abstract class XLShape	

Remarks

Excel sheets may contain many types of embedded shapes, including images, graphical elements, controls, and comments.

The **XLShape** abstract class contains information that is common to all shape types, including the shape's location ([Sheet](#), [Row](#), [Column](#), [Rectangle](#), [Rotation](#)), the type of border drawn around the shape ([LineWidth](#), [LineColor](#), [LineStyle](#)), the [Rotation](#) applied to the shape, and [Hyperlink](#) information.

The [XLPictureShape](#) class derives from **XLShape** and is used to embed images in sheets.

Inheritance Hierarchy

System.Object

C1.C1Excel.XLShape

[C1.C1Excel.XLOpaqueShape](#)

[C1.C1Excel.XLPictureShape](#)

[C1.C1Excel.XLTextShape](#)

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Members](#)












[C1.C1Excel Namespace](#)



Members

[Properties](#) [Methods](#)

The following tables list the members exposed by [XLShape](#).


Public Properties

	Name	Description
	Column	Gets the index of the column to which the shape is attached.
	DashedLineStyle	Gets or sets the dash style of the line or border around the shape.
	Hyperlink	Gets or sets the hyperlink associated with the shape.
	Id	Gets the unique identifier of the shape.
	IsEmpty	Determines whether the shape is empty.
	LineColor	Gets or sets the color of the border around the shape.
	LineStyle	Gets or sets the style of the line or border around the shape.
	LineWidth	Gets or sets the width of the border around the shape, in twips.
	Rectangle	Gets or sets the rectangle that contains the shape, in twips.
	Rotation	Gets or sets the rotation of the shape, in degrees.
	Row	Gets the index of the row to which the shape is attached.

	Sheet	Gets a reference to the parent XLSheet object.
	Workbook	Gets a reference to the parent C1XLBook object.

[Top](#)

Public Methods

	Name	Description
	Clone	Creates a new XLShape object that is a copy of the current instance.

[Top](#)

See Also

Reference


[XLShape Class](#)

[C1.C1Excel Namespace](#)

Methods

For a list of all members of this type, see [XLShape members](#).

Public Methods

	Name	Description
	Clone	Creates a new XLShape object that is a copy of the current instance.

[Top](#)

See Also

Reference

[XLShape Class](#)

[C1.C1Excel Namespace](#)

Clone Method

Creates a new [XLShape](#) object that is a copy of the current instance.

Syntax

Visual Basic (Declaration)	
Public Function Clone() As XLShape	
C#	
public XLShape Clone()	

Return Value

A new [XLShape](#) object that is a copy of the current instance.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference





[XLShape Class](#)










[XLShape Members](#)

Properties

For a list of all members of this type, see [XLShape members](#).

Public Properties

	Name	Description
	Column	Gets the index of the column to which the shape is attached.
	DashedLineStyle	Gets or sets the dash style of the line or border around the shape.
	Hyperlink	Gets or sets the hyperlink associated with the shape.
	Id	Gets the unique identifier of the shape.

	IsEmpty	Determines whether the shape is empty.
	LineColor	Gets or sets the color of the border around the shape.
	LineStyle	Gets or sets the style of the line or border around the shape.
	LineWidth	Gets or sets the width of the border around the shape, in twips.
	Rectangle	Gets or sets the rectangle that contains the shape, in twips.
	Rotation	Gets or sets the rotation of the shape, in degrees.
	Row	Gets the index of the row to which the shape is attached.
	Sheet	Gets a reference to the parent XLSheet object.
	Workbook	Gets a reference to the parent C1XLBook object.

[Top](#)

See Also

Reference

[XLShape Class](#)

[C1.C1Excel Namespace](#)

Column Property

Gets the index of the column to which the shape is attached.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Column As System.Integer</code>	
C#	
<code>public System.int Column {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Class](#)

[XLShape Members](#)

DashedLineStyle Property

Gets or sets the dash style of the line or border around the shape.

Syntax

Visual Basic (Declaration)	
<code>Public Property DashedLineStyle As XLShapeDashedLineStyleEnum</code>	
C#	
<code>public XLShapeDashedLineStyleEnum DashedLineStyle {get; set;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Class](#)

[XLShape Members](#)

Hyperlink Property

Gets or sets the hyperlink associated with the shape.

Syntax

Visual Basic (Declaration)	
----------------------------	--

<code>Public Property Hyperlink As System.String</code>
C#
<code>public System.string Hyperlink {get; set;}</code>

Remarks

If you set this property to a URL, clicking the shape in Excel will open the browser and navigate to the URL.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Class](#)

[XLShape Members](#)

Id Property

Gets the unique identifier of the shape.

Syntax

Visual Basic (Declaration)
<code>Public ReadOnly Property Id As System.Integer</code>
C#
<code>public System.int Id {get;}</code>

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Class](#)

[XLShape Members](#)

IsEmpty Property

Determines whether the shape is empty.

Syntax

Visual Basic (Declaration)	
<code>Public MustOverride ReadOnly Property IsEmpty As System.Boolean</code>	
C#	
<code>public abstract System.bool IsEmpty {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Class](#)

[XLShape Members](#)

LineColor Property

Gets or sets the color of the border around the shape.

Syntax

Visual Basic (Declaration)	
<code>Public Property LineColor As System.Drawing.Color</code>	
C#	
<code>public System.Drawing.Color LineColor {get; set;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Class](#)

[XLShape Members](#)

LineStyle Property

Gets or sets the style of the line or border around the shape.

Syntax

Visual Basic (Declaration)	
<code>Public Property LineStyle As XLShapeLineStyleEnum</code>	
C#	
<code>public XLShapeLineStyleEnum LineStyle {get; set;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Class](#)

[XLShape Members](#)

LineWidth Property

Gets or sets the width of the border around the shape, in twips.

Syntax

Visual Basic (Declaration)	
----------------------------	--

Public Property LineWidth As System.Single
C#
public System.float LineWidth {get; set;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Class](#)

[XLShape Members](#)

Rectangle Property

Gets or sets the rectangle that contains the shape, in twips.

Syntax

Visual Basic (Declaration)
Public Property Rectangle As System.Drawing.Rectangle
C#
public System.Drawing.Rectangle Rectangle {get; set;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Class](#)

[XLShape Members](#)

Rotation Property

Gets or sets the rotation of the shape, in degrees.

Syntax

Visual Basic (Declaration)

```
Public Property Rotation As System.Single
```

C#

```
public System.float Rotation {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Class](#)

[XLShape Members](#)

Row Property

Gets the index of the row to which the shape is attached.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property Row As System.Integer
```

C#

```
public System.int Row {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Class](#)

[XLShape Members](#)

Sheet Property

Gets a reference to the parent [XLSheet](#) object.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Sheet As XLSheet</code>	
C#	
<code>public XLSheet Sheet {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Class](#)

[XLShape Members](#)

Workbook Property

Gets a reference to the parent [C1XLBook](#) object.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Workbook As C1XLBook</code>	
C#	

```
public C1XLBook Workbook {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLShape Class](#)

[XLShape Members](#)

XLSheet

[Example](#)

Represents individual worksheets in an Excel workbook ([C1XLBook](#)).

Object Model

XLSheet

Syntax

Visual Basic (Declaration)	
<pre><System.Reflection.DefaultMemberAttribute("Item")> Public Class XLSheet</pre>	
C#	
<pre>[System.Reflection.DefaultMemberAttribute("Item")] public class XLSheet</pre>	

Remarks

Provides indexers to get or set the value of individual cells ([XLCell](#)) and to access the [Rows](#) and [Columns](#) on the sheet.

Example

Use the [C1XLBook.C1XLBook.Sheets](#) indexer to get an individual sheet. For example, the code below gets a reference to the first sheet on the book, then prints the number of rows and columns on the sheet:

- [C#](#)

```
C1XLBook book = new C1XLBook();
XLSheet sheet = book.Sheets[0];
Debug.WriteLine("Sheet has {0} rows and {1} columns",
    sheet.Rows.Count, sheet.Columns.Count);
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLSheet

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Members](#)

[C1.C1Excel Namespace](#)

[Overview](#)

[Example](#)

Represents individual worksheets in an Excel workbook ([C1XLBook](#)).

Object Model

XLSheet

Syntax

Visual Basic (Declaration)

```
<System.Reflection.DefaultMemberAttribute("Item")>
Public Class XLSheet
```

C#

```
[System.Reflection.DefaultMemberAttribute("Item")]  
public class XLSheet
```

Remarks

Provides indexers to get or set the value of individual cells ([XLCell](#)) and to access the [Rows](#) and [Columns](#) on the sheet.

Example

Use the [C1XLBook.C1XLBook.Sheets](#) indexer to get an individual sheet. For example, the code below gets a reference to the first sheet on the book, then prints the number of rows and columns on the sheet:

- [C#](#)

```
C1XLBook book = new C1XLBook();  
XLSheet sheet = book.Sheets[0];  
Debug.WriteLine("Sheet has {0} rows and {1} columns",  
    sheet.Rows.Count, sheet.Columns.Count);
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLSheet

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Members](#)















[C1.C1Excel Namespace](#)













Members

[Properties](#) [Methods](#)

The following tables list the members exposed by [XLSheet](#).


Public Properties









	Name	Description
	Book	Gets a reference to the C1XLBook that owns the sheet.
	ColumnMaxLevel	Gets maximum outline level for columns.
	Columns	Gets a reference to sheet's column collection.
	Comments	Gets an XLCommentCollection that contains the collection of comments that on the sheet.
	DefaultColumnWidth	Gets or sets the default column width for the sheet (in twips).
	DefaultRowHeight	Gets or sets the default row height for the sheet (in twips).
	GridColor	Gets or sets the color used to display gridlines.
	Item	Gets the cell at a specified position on the sheet, creating a new cell if necessary.
	Locked	Gets or sets a value that determines if the sheet is locked for editing.
	MaxOutlineLevel	Gets the maximum subtotals outline level for rows or columns.
	MergedCells	Gets an XLCellRangeCollection that contains the collection of cells that are merged on the sheet.
	Name	Gets or sets the name of the XLSheet .
	OutlinesBelow	Gets or sets a value indicating the vertical location of outline buttons.
	OutlinesRight	Gets or sets a value indicating the horizontal location of outline buttons.

	PrintSettings	Gets or sets an XLPrintSettings object that controls how the sheet is printed.
	RowMaxLevel	Gets maximum outline level for rows.
	Rows	Gets a reference to the sheet's row collection.
	Scale	Gets or sets the view scale of this worksheet using percentages.
	SelectedCells	Gets an XLCellRangeCollection that contains the collection of cells that are selected on the sheet.
	Shapes	Gets a reference to the ShapeCollection for the sheet.
	ShowGridLines	Gets or sets whether Excel should show the grid lines when displaying the sheet.
	ShowHeaders	Gets or sets whether Excel should show the row and column headers when displaying the sheet.
	ShowZeros	Gets or sets whether Excel should show the zero values on the sheet.
	TabColor	Gets or sets the color used to display the tab of this sheet.
	TotalsBelowData	Gets whether Excel should show the subtotals data when displaying the sheet.
	Visible	Gets or sets the sheet's visibility.

[Top](#)

Public Methods

	Name	Description
	Clone	Creates a copy of this sheet.

⇒  CopyFormula	Copies the formula from a source cell to a destination cell, adjusting relative references.
⇒  GetCell	Gets a reference to a cell at the specified coordinates or null if the cell doesn't exist.
⇒  GetFormattedText	Gets formatted text is using XLStyle format value of the cell.
⇒  GetRangeToRepeat	Returns a value indicating whether print titles are specified for this worksheet. (Print titles are rows repeated at top, and columns repeated at left of each page when the sheet is printed.) Output parameters indicate the indices of title rows and columns.
⇒  Load	Overloaded. Loads the Excel worksheet from a file.
⇒  LoadCsv	Overloaded. Loads data from a file containing comma-separated values (.csv) into the current sheet.
⇒  SaveCsv	Overloaded. Saves data from the current sheet into a comma-separated values (.csv) formatted file.
⇒  SetRangeToRepeat	Specifies the range of rows and columns to use as print titles for this worksheet. (Print titles are rows repeated at top, and columns repeated at left of each page when the sheet is printed.)

[Top](#)

See Also

Reference

[XLSheet Class](#)

[C1.C1Excel Namespace](#)

Methods

For a list of all members of this type, see [XLSheet members](#).

Public Methods

	Name	Description
≡	Clone	Creates a copy of this sheet.
≡	CopyFormula	Copies the formula from a source cell to a destination cell, adjusting relative references.
≡	GetCell	Gets a reference to a cell at the specified coordinates or null if the cell doesn't exist.
≡	GetFormattedText	Gets formatted text is using XLStyle format value of the cell.
≡	GetRangeToRepeat	Returns a value indicating whether print titles are specified for this worksheet. (Print titles are rows repeated at top, and columns repeated at left of each page when the sheet is printed.) Output parameters indicate the indices of title rows and columns.
≡	Load	Overloaded. Loads the Excel worksheet from a file.
≡	LoadCsv	Overloaded. Loads data from a file containing comma-separated values (.csv) into the current sheet.
≡	SaveCsv	Overloaded. Saves data from the current sheet into a comma-separated values (.csv) formatted file.
≡	SetRangeToRepeat	Specifies the range of rows and columns to use as print titles for this worksheet. (Print titles are rows repeated at top, and columns repeated at left of each page when the sheet is printed.)

[Top](#)

See Also

Reference

[XLSheet Class](#)

[C1.C1Excel Namespace](#)

Clone Method

[Example](#)

Creates a copy of this sheet.

Syntax

Visual Basic (Declaration)

```
Public Function Clone() As XLSheet
```

C#

```
public XLSheet Clone()
```

Return Value

A new [XLSheet](#) object with the same contents and formatting as this sheet.

Remarks

After cloning a sheet, you must rename it and then add it to the book (duplicate names are not allowed).

This method is useful for applications that generate books with a large number of similar sheets.

Example

The code below loads a book that contains a template sheet, creates 12 copies of that sheet, removes the template sheet, then saves the file with a new name.

- [C#](#)

```
// load book with template sheet
_c1xl.Load(@"c:\temp\template.xls");

// create 12 copies of the template sheet
XLSheet templateSheet = _c1xl.Sheets["Template"];
for (int month = 1; month <= 12; month++)
{
    XLSheet newSheet = templateSheet.Clone();
    newSheet.Name = month.ToString(); // rename clone
    newSheet[0,0].Value = month;      // make changes
    _c1xl.Sheets.Add(newSheet);        // add clone to book
}

// remove the template sheet and save with new name
_c1xl.Sheets.Remove("Template");
```

```
_c1xl.Save(@"C:\temp\expense_report.xls");
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

CopyFormula Method

The row index of a source cell with exist formula.

The column index of a source cell with exist formula.

The row index of a destination cell for copies the formula.

The column index of a destination cell for copies the formula.

Copies the formula from a source cell to a destination cell, adjusting relative references.

Syntax

Visual Basic (Declaration)

```
Public Sub CopyFormula( _  
    ByVal rowFrom As System.Integer, _  
    ByVal colFrom As System.Integer, _  
    ByVal rowTo As System.Integer, _  
    ByVal colTo As System.Integer _  
)
```

C#

```
public void CopyFormula(  
    System.int rowFrom,  
    System.int colFrom,  
    System.int rowTo,  
    System.int colTo
```

)

Parameters

rowFrom

The row index of a source cell with exist formula.

colFrom

The column index of a source cell with exist formula.

rowTo

The row index of a destination cell for copies the formula.

colTo

The column index of a destination cell for copies the formula.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

GetCell Method

Index of the row.

Index of the column.

Gets a reference to a cell at the specified coordinates or null if the cell doesn't exist.

Syntax

Visual Basic (Declaration)

```
Public Function GetCell( _  
    ByVal rowIndex As System.Integer, _
```

```
ByVal colIndex As System.Integer _  
) As XLCeIl
```

C#

```
public XLCeIl GetCell(  
    System.int rowIndex,  
    System.int colIndex  
)
```

Parameters

rowIndex

Index of the row.

colIndex

Index of the column.

Return Value

A reference to the [XLCeIl](#) object at the specified coordinates, or null if there is no cell at the specified position.

Remarks

To populate new sheets, use the [C1XLBook.Sheets](#) indexer instead.

The indexer will automatically create new rows, columns, and cells as needed, and will never return null.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

GetFormattedText Method

Index of the row.

Index of the column.

Gets formatted text is using [XLStyle](#) format value of the cell.

Syntax

Visual Basic (Declaration)	
<pre>Public Function GetFormattedText(_ ByVal rowIndex As System.Integer, _ ByVal colIndex As System.Integer _) As System.String</pre>	
C#	
<pre>public System.string GetFormattedText(System.int rowIndex, System.int colIndex)</pre>	

Parameters

rowIndex

Index of the row.

colIndex

Index of the column.

Return Value

The formatted text of the cell with row and column indexes.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)
[XLSheet Members](#)

GetRangeToRepeat Method

OUT: index of the first title row (repeated at top).

OUT: index of the last title row (repeated at top).

OUT: index of the first title column (repeated at left).

OUT: index of the last title column (repeated at left).

Returns a value indicating whether print titles are specified for this worksheet. (Print titles are rows repeated at top, and columns repeated at left of each page when the sheet is printed.) Output parameters indicate the indices of title rows and columns.

Syntax

Visual Basic (Declaration)

```
Public Function GetRangeToRepeat( _  
    ByRef rowFrom As System.Integer, _  
    ByRef rowTo As System.Integer, _  
    ByRef colFrom As System.Integer, _  
    ByRef colTo As System.Integer _  
) As System.Boolean
```

C#

```
public System.bool GetRangeToRepeat(  
    out System.int rowFrom,  
    out System.int rowTo,  
    out System.int colFrom,  
    out System.int colTo  
)
```

Parameters

rowFrom

OUT: index of the first title row (repeated at top).

rowTo

OUT: index of the last title row (repeated at top).

colFrom

OUT: index of the first title column (repeated at left).

colTo

OUT: index of the last title column (repeated at left).

Return Value

True if title rows or columns are specified for this sheet, false otherwise.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

Load Method

Loads the Excel worksheet from a file.

Overload List

Overload	Description
Load(String,Boolean)	Loads the Excel worksheet from a file.
Load(Stream,Boolean)	Loads the Excel worksheet from a stream.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

Load(String, Boolean) Method

Name of the file that contains the worksheet.

True to finish loading from the workbook; **False** to load data into the other worksheets.

Loads the Excel worksheet from a file.

Syntax

Visual Basic (Declaration)

```
Public Overloads Sub Load( _  
    ByVal fileName As System.String, _  
    ByVal last As System.Boolean _  
)
```

C#

```
public void Load(  
    System.string fileName,  
    System.bool last  
)
```

Parameters

fileName

Name of the file that contains the worksheet.

last

True to finish loading from the workbook; **False** to load data into the other worksheets.

Remarks

It was previously necessary to load the workbook from the stream without filling the worksheets.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)
[XLSheet Members](#)
[Overload List](#)

Load(Stream, Boolean) Method

System.IO.Stream that contains the worksheet.

True to finish loading from the workbook; **False** to load data into the other worksheets.

Loads the Excel worksheet from a stream.

Syntax

Visual Basic (Declaration)

```
Public Overloads Sub Load( _  
    ByVal stream As System.IO.Stream, _  
    ByVal last As System.Boolean _  
)
```

C#

```
public void Load(  
    System.IO.Stream stream,  
    System.bool last  
)
```

Parameters

stream

System.IO.Stream that contains the worksheet.

last

True to finish loading from the workbook; **False** to load data into the other worksheets.

Remarks

It was previously necessary to load the workbook from the stream without filling the worksheets.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)
[XLSheet Members](#)
[Overload List](#)

LoadCsv Method

Loads data from a file containing comma-separated values (.csv) into the current sheet.

Overload List

Overload	Description
LoadCsv(String)	Loads data from a file containing comma-separated values (.csv) into the current sheet.
LoadCsv(Stream)	Loads data from a stream containing comma-separated values (.csv) into the current sheet.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

LoadCsv(String) Method

The name of a file containing data in .csv format.

Loads data from a file containing comma-separated values (.csv) into the current sheet.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Sub LoadCsv(_ ByVal fileName As System.String _)</pre>	
C#	
<pre>public void LoadCsv(System.string fileName)</pre>	

Parameters

fileName

The name of a file containing data in .csv format.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

[Overload List](#)

LoadCsv(Stream) Method

A stream containing data in .csv format.

Loads data from a stream containing comma-separated values (.csv) into the current sheet.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Sub LoadCsv(_ ByVal stream As System.IO.Stream _)</pre>	
C#	
<pre>public void LoadCsv(System.IO.Stream stream)</pre>	

Parameters

stream

A stream containing data in .csv format.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

[Overload List](#)

SaveCsv Method

Saves data from the current sheet into a comma-separated values (.csv) formatted file.

Overload List

Overload	Description
SaveCsv(String)	Saves data from the current sheet into a comma-separated values (.csv)

	formatted file.
SaveCsv(Stream)	Saves data from the current sheet into a comma-separated values (.csv) formatted stream.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

SaveCsv(String) Method

The target (.csv) file name.

Saves data from the current sheet into a comma-separated values (.csv) formatted file.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Sub SaveCsv(_ ByVal fileName As System.String _)</pre>	
C#	
<pre>public void SaveCsv(System.string fileName)</pre>	

Parameters

fileName

The target (.csv) file name.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

[Overload List](#)

SaveCsv(Stream) Method

The target stream.

Saves data from the current sheet into a comma-separated values (.csv) formatted stream.

Syntax

Visual Basic (Declaration)

```
Public Overloads Sub SaveCsv( _  
    ByVal stream As System.IO.Stream _  
)
```

C#

```
public void SaveCsv(  
    System.IO.Stream stream  
)
```

Parameters

stream

The target stream.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

[Overload List](#)

SetRangeToRepeat Method

Index of the first title row (repeated at top).

Index of the last title row (repeated at top).

Index of the first title column (repeated at left).

Index of the last title column (repeated at left).

Specifies the range of rows and columns to use as print titles for this worksheet. (Print titles are rows repeated at top, and columns repeated at left of each page when the sheet is printed.)

Syntax

Visual Basic (Declaration)

```
Public Sub SetRangeToRepeat( _  
    ByVal rowFrom As System.Integer, _  
    ByVal rowTo As System.Integer, _  
    ByVal colFrom As System.Integer, _  
    ByVal colTo As System.Integer _  
)
```

C#

```
public void SetRangeToRepeat(  
    System.int rowFrom,  
    System.int rowTo,  
    System.int colFrom,  
    System.int colTo  
)
```

Parameters

rowFrom

Index of the first title row (repeated at top).

rowTo

Index of the last title row (repeated at top).

colFrom

Index of the first title column (repeated at left).

colTo

Index of the last title column (repeated at left).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference






[XLSheet Class](#)
















[XLSheet Members](#)







Properties

For a list of all members of this type, see [XLSheet members](#).

Public Properties

	Name	Description
	Book	Gets a reference to the C1XLBook that owns the sheet.
	ColumnMaxLevel	Gets maximum outline level for columns.
	Columns	Gets a reference to sheet's column collection.
	Comments	Gets an XLCommentCollection that contains the collection of comments that on the sheet.
	DefaultColumnWidth	Gets or sets the default column width for the sheet (in twips).

 DefaultRowHeight	Gets or sets the default row height for the sheet (in twips).
 GridColor	Gets or sets the color used to display gridlines.
 Item	Gets the cell at a specified position on the sheet, creating a new cell if necessary.
 Locked	Gets or sets a value that determines if the sheet is locked for editing.
 MaxOutlineLevel	Gets the maximum subtotals outline level for rows or columns.
 MergedCells	Gets an XLCellRangeCollection that contains the collection of cells that are merged on the sheet.
 Name	Gets or sets the name of the XLSheet .
 OutlinesBelow	Gets or sets a value indicating the vertical location of outline buttons.
 OutlinesRight	Gets or sets a value indicating the horizontal location of outline buttons.
 PrintSettings	Gets or sets an XLPrintSettings object that controls how the sheet is printed.
 RowMaxLevel	Gets maximum outline level for rows.
 Rows	Gets a reference to the sheet's row collection.
 Scale	Gets or sets the view scale of this worksheet using percentages.
 SelectedCells	Gets an XLCellRangeCollection that contains the collection of cells that are selected on the sheet.
 Shapes	Gets a reference to the ShapeCollection for the sheet.

	ShowGridLines	Gets or sets whether Excel should show the grid lines when displaying the sheet.
	ShowHeaders	Gets or sets whether Excel should show the row and column headers when displaying the sheet.
	ShowZeros	Gets or sets whether Excel should show the zero values on the sheet.
	TabColor	Gets or sets the color used to display the tab of this sheet.
	TotalsBelowData	Gets whether Excel should show the subtotals data when displaying the sheet.
	Visible	Gets or sets the sheet's visibility.

[Top](#)

See Also

Reference

[XLSheet Class](#)

[C1.C1Excel Namespace](#)

Book Property

Gets a reference to the [C1XLBook](#) that owns the sheet.

Syntax

Visual Basic (Declaration)	
Public ReadOnly Property Book As C1XLBook	
C#	
public C1XLBook Book { get ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

ColumnMaxLevel Property

Gets maximum outline level for columns.

Syntax

Visual Basic (Declaration)	
Public ReadOnly Property ColumnMaxLevel As System.Integer	
C#	
public System.int ColumnMaxLevel { get ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

Columns Property

Gets a reference to sheet's column collection.

Syntax

Visual Basic (Declaration)	
----------------------------	--

Public ReadOnly Property Columns As XLColumnCollection
C#
public XLColumnCollection Columns {get;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

Comments Property

Gets an [XLCommentCollection](#) that contains the collection of comments that on the sheet.

Syntax

Visual Basic (Declaration)
Public ReadOnly Property Comments As XLCommentCollection
C#
public XLCommentCollection Comments {get;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

DefaultColumnWidth Property

Gets or sets the default column width for the sheet (in twips).

Syntax

Visual Basic (Declaration)

```
Public Property DefaultColumnWidth As System.Integer
```

C#

```
public System.Int DefaultColumnWidth {get; set;}
```

Remarks

You can set the width of individual rows using the [XLColumn](#) class. Any columns that do not have a custom width assigned to them will be displayed using the sheet's [DefaultColumnWidth](#).

The [DefaultColumnWidth](#) property is expressed in twips (1/20th of a point), rather than pixels. This allows sheets to maintain their aspect regardless of the resolution of the display.

To convert twips into pixels, use the [C1XLBook.TwipsToPixels](#) method in [C1XLBook](#). To convert pixels into twips, use the [PixelsToTwips](#) method.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

DefaultRowHeight Property

Gets or sets the default row height for the sheet (in twips).

Syntax

Visual Basic (Declaration)

Public Property DefaultRowHeight As System.Integer
C#
public System.Integer DefaultRowHeight {get; set;}

Remarks

You can set the height of individual rows using the [XLRow](#) class. Any rows that do not have a custom height assigned to them will be displayed using the sheet's [DefaultRowHeight](#).

The [DefaultRowHeight](#) property is expressed in twips (1/20th of a point), rather than pixels. This allows sheets to maintain their aspect regardless of the resolution of the display.

To convert twips into pixels, use the [C1XLBook.TwipsToPixels](#) method in [C1XLBook](#). To convert pixels into twips, use the [PixelsToTwips](#) method.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

GridColor Property

Gets or sets the color used to display gridlines.

Syntax

Visual Basic (Declaration)
Public Property GridColor As System.Drawing.Color
C#
public System.Drawing.Color GridColor {get; set;}

Remarks

Set this property to **Color.Transparent** to display the grid lines using the default color.

To hide the grid lines, set the [ShowGridLines](#) property to false.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

Item Property

Gets the cell at a specified position on the sheet, creating a new cell if necessary.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Default Property Item( _  
    ByVal rowIndex As System.Integer, _  
    ByVal colIndex As System.Integer _  
) As XLCell
```

C#

```
public XLCell this[  
    System.int rowIndex,  
    System.int colIndex  
]; {get;}
```

Parameters

rowIndex

colIndex

Remarks

If the specified cell doesn't exist when the indexer is invoked, the sheet will be expanded and a new cell will be created, then returned. This makes it easy to create and populate sheets.

If you want to determine whether a specific cell has been defined, use the [GetCell](#) method instead.

The indexer is generally more useful when creating and populating sheets. The [GetCell](#) method is more useful when loading existing sheets.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)
[XLSheet Members](#)

Locked Property
[Example](#)

Gets or sets a value that determines if the sheet is locked for editing.

Syntax

Visual Basic (Declaration)	
Public Property Locked As System.Boolean	
C#	
public System.bool Locked { get ; set ;}	

Remarks

Sheets and styles can be locked. By default, sheets are unlocked and styles are locked. This combination allows users to edit the cells in Excel.

To protect a cell against editing in Excel, both the sheet and the cell style must have the [Locked](#) property set to true.

To lock most cells on a sheet and allow editing of only a few cells, lock the sheet, then create an unlocked style and assign it to the cells that should be editable.

Example

The code below creates a data entry sheet. Most cells are locked, except for the ones where the user is supposed to enter data.

- [C#](#)

```
// start with a single locked sheet
_c1xl.Clear();
XLSheet sheet = _c1xl.Sheets[0];
sheet.Locked = true;

// create an unlocked style
XLStyle dataEntry = new XLStyle(_c1xl);
dataEntry.Locked = false;
dataEntry.BackColor = Color.Beige;

// create data entry titles
sheet[0,0].Value = "Name:";
sheet[1,0].Value = "Address:";
sheet[2,0].Value = "Phone #:";

create data entry cells (unlocked)
sheet[0,1].Style = dataEntry;
sheet[1,1].Style = dataEntry;
sheet[2,1].Style = dataEntry;

// save the book
_c1xl.Save(@"c:\temp\Protected.xls");
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

MaxOutlineLevel Property

Gets the maximum subtotals outline level for rows or columns.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property MaxOutlineLevel As System.Integer</code>	
C#	
<code>public System.int MaxOutlineLevel {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

MergedCells Property

Gets an [XLCellRangeCollection](#) that contains the collection of cells that are merged on the sheet.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property MergedCells As XLCellRangeCollection</code>	
C#	
<code>public XLCellRangeCollection MergedCells {get;}</code>	

Remarks

The collection has methods for inspecting, adding, or clearing merged ranges in a sheet. Each merged range is represented by an [XLCellRange](#) object.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

Name Property

Gets or sets the name of the [XLSheet](#).

Syntax

Visual Basic (Declaration)	
Public Property Name As System.String	
C#	
public System.string Name { get ; set ;}	

Remarks

When you open a workbook in Excel, the sheet names appear in the tabs below the work area.

Sheet names can be used as indexers, so they should be unique.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

OutlinesBelow Property

Gets or sets a value indicating the vertical location of outline buttons.

Syntax

Visual Basic (Declaration)	
Public Property OutlinesBelow As System.Boolean	
C#	
public System.bool OutlinesBelow { get ; set ;}	

Remarks

If this property is set to true, outline buttons are located below the outline group, otherwise they are located above the outline group.

The default value is **true**

.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

OutlinesRight Property

Gets or sets a value indicating the horizontal location of outline buttons.

Syntax

Visual Basic (Declaration)	
Public Property OutlinesRight As System.Boolean	
C#	
public System.bool OutlinesRight { get ; set ;}	

Remarks

If this property is set to true, outline buttons are located to the right the outline group, otherwise they are located to the left of the outline group.

The default value is **true**

.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

PrintSettings Property

[Example](#)

Gets or sets an [XLPrintSettings](#) object that controls how the sheet is printed.

Syntax

Visual Basic (Declaration)	
Public Property PrintSettings As XLPrintSettings	
C#	
public XLPrintSettings PrintSettings { get ; set ;}	

Example

The code below creates a header for the sheet and sets the orientation to landscape:

- [C#](#)

```
XLPrintSettings pp = sheet.PrintSettings();  
pp.Landscape = true;  
pp.Header = "&LLeft Header&CCenter Header&RRight Header";
```


Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

RowMaxLevel Property

Gets maximum outline level for rows.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property RowMaxLevel As System.Integer
```

C#

```
public System.int RowMaxLevel {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

Rows Property

Gets a reference to the sheet's row collection.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Rows As XLRowCollection</code>	
C#	
<code>public XLRowCollection Rows {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

Scale Property

Gets or sets the view scale of this worksheet using percentages.

Syntax

Visual Basic (Declaration)	
<code>Public Property Scale As System.Integer</code>	
C#	
<code>public System.int Scale {get; set;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

SelectedCells Property

Gets an [XLCellRangeCollection](#) that contains the collection of cells that are selected on the sheet.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property SelectedCells As XLCellRangeCollection
```

C#

```
public XLCellRangeCollection SelectedCells {get;}
```

Remarks

The collection has methods for inspecting, adding, or clearing merged ranges in a sheet. Each selected range is represented by an [XLCellRange](#) object.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

Shapes Property

Gets a reference to the [ShapeCollection](#) for the sheet.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property Shapes As ShapeCollection
```

C#

```
public ShapeCollection Shapes {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

ShowGridLines Property

Gets or sets whether Excel should show the grid lines when displaying the sheet.

Syntax

Visual Basic (Declaration)

```
Public Property ShowGridLines As System.Boolean
```

C#

```
public System.bool ShowGridLines {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

ShowHeaders Property

Gets or sets whether Excel should show the row and column headers when displaying the sheet.

Syntax

Visual Basic (Declaration)	
Public Property ShowHeaders As System.Boolean	
C#	
public System.bool ShowHeaders { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

ShowZeros Property

Gets or sets whether Excel should show the zero values on the sheet.

Syntax

Visual Basic (Declaration)	
Public Property ShowZeros As System.Boolean	
C#	
public System.bool ShowZeros { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

TabColor Property

Gets or sets the color used to display the tab of this sheet.

Syntax

Visual Basic (Declaration)

```
Public Property TabColor As System.Drawing.Color
```

C#

```
public System.Drawing.Color TabColor {get; set;}
```

Remarks

Set this property to **Color.Transparent** to display the sheet tab using the default color.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

TotalsBelowData Property

Gets whether Excel should show the subtotals data when displaying the sheet.

Syntax

Visual Basic (Declaration)

```
<System.ObsoleteAttribute(>
```

```
Public ReadOnly Property TotalsBelowData As System.Boolean
```

C#

```
[System.ObsoleteAttribute()]  
public System.bool TotalsBelowData {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

Visible Property

Gets or sets the sheet's visibility.

Syntax

Visual Basic (Declaration)

```
Public Property Visible As System.Boolean
```

C#

```
public System.bool Visible {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheet Class](#)

[XLSheet Members](#)

XLSheetCollection

[Example](#)

Collection of [XLSheet](#) objects that represent the individual worksheets in a [C1XLBook](#).

Object Model

XLSheetCollection

Syntax

Visual Basic (Declaration)	
<pre><System.Reflection.DefaultMemberAttribute("Item")> Public Class XLSheetCollection</pre>	
C#	
<pre>[System.Reflection.DefaultMemberAttribute("Item")] public class XLSheetCollection</pre>	

Remarks

Provides methods and properties for counting, enumerating, adding, and removing sheets from the workbook.

Example

Use the [C1XLBook.C1XLBook.Sheets](#) property to get the book's sheet collection. For example, the code below gets a reference to the first sheet on the book and then prints the sheet's name:

- [C#](#)

```
C1XLBook book = new C1XLBook();
XLSheet sheet = book.Sheets[0];
Debug.WriteLine(sheet.Name);
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLSheetCollection

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Members](#)

[C1.C1Excel Namespace](#)

[Overview](#)

[Example](#)

Collection of [XLSheet](#) objects that represent the individual worksheets in a [C1XLBook](#).

Object Model

XLSheetCollection

Syntax

Visual Basic (Declaration)

```
<System.Reflection.DefaultMemberAttribute("Item")>  
Public Class XLSheetCollection
```

C#

```
[System.Reflection.DefaultMemberAttribute("Item")]  
public class XLSheetCollection
```

Remarks

Provides methods and properties for counting, enumerating, adding, and removing sheets from the workbook.

Example

Use the [C1XLBook.C1XLBook.Sheets](#) property to get the book's sheet collection. For example, the code below gets a reference to the first sheet on the book and then prints the sheet's name:

- [C#](#)

```
C1XLBook book = new C1XLBook();  
XLSheet sheet = book.Sheets[0];  
Debug.WriteLine(sheet.Name);
```

Inheritance Hierarchy

System.Object

C1.C1Excel.XLSheetCollection

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Members](#)






[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)

The following tables list the members exposed by [XLSheetCollection](#).

Public Properties

	Name	Description
	Book	Gets a reference to the C1XLBook that owns the collection.
	Count	Gets the number of XLSheet objects in the collection.
	FirstIndex	Gets or sets the index of the sheet that is initially opened when a C1XLBook file is loaded into Excel.
	Item	Overloaded. Gets the XLSheet at a given position in the collection.
	SelectedIndex	Gets or sets the index of the sheet that is selected when a C1XLBook file is loaded into Excel.

[Top](#)

Public Methods

	Name	Description
≡	Add	Overloaded. Creates a new XLSheet and appends it to the collection.
≡	Clear	Removes all items from the collection.
≡	Contains	Overloaded. Determines whether the collection contains a specific XLSheet object.
≡	IndexOf	Overloaded. Gets the index of a given XLSheet in the collection.
≡	Insert	Overloaded. Creates a new XLSheet and inserts it at a specific position in the collection.
≡	Remove	Overloaded. Removes an XLSheet from the collection.
≡	RemoveAt	Removes the XLSheet at a specific index from the collection.

[Top](#)

See Also

Reference

[XLSheetCollection Class](#)

[C1.C1Excel Namespace](#)

Methods

For a list of all members of this type, see [XLSheetCollection members](#).

Public Methods

	Name	Description
≡	Add	Overloaded. Creates a new XLSheet and appends it to the collection.
≡	Clear	Removes all items from the collection.

≡	Contains	Overloaded. Determines whether the collection contains a specific XLSheet object.
≡	IndexOf	Overloaded. Gets the index of a given XLSheet in the collection.
≡	Insert	Overloaded. Creates a new XLSheet and inserts it at a specific position in the collection.
≡	Remove	Overloaded. Removes an XLSheet from the collection.
≡	RemoveAt	Removes the XLSheet at a specific index from the collection.

[Top](#)

See Also

Reference

[XLSheetCollection Class](#)

[C1.C1Excel Namespace](#)

Add Method

Creates a new [XLSheet](#) and appends it to the collection.

Overload List

Overload	Description
Add()	Creates a new XLSheet and appends it to the collection.
Add(XLSheet)	Appends an existing XLSheet to the collection.
Add(String)	Creates a new XLSheet with a given name and appends it to the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

Add() Method

Creates a new [XLSheet](#) and appends it to the collection.

Syntax

Visual Basic (Declaration)	
Public Overloads Function Add() As XLSheet	
C#	
public XLSheet Add()	

Return Value

A reference to the new [XLSheet](#) object.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

[Overload List](#)

Add(XLSheet) Method

The object to add to the collection.

Appends an existing [XLSheet](#) to the collection.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Add( _  
    ByVal sheet As XLSheet _  
) As XLSheet
```

C#

```
public XLSheet Add(  
    XLSheet sheet  
)
```

Parameters

sheet

The object to add to the collection.

Return Value

A reference to the object that was added to the collection (in this case, always the [sheet](#) parameter).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)
[XLSheetCollection Members](#)
[Overload List](#)

Add(String) Method

The name of the new [XLSheet](#).

Creates a new [XLSheet](#) with a given name and appends it to the collection.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Add( _  
    ByVal name As System.String _  
) As XLSheet
```

C#

```
public XLSheet Add(  
    System.string name  
)
```

Parameters

name

The name of the new [XLSheet](#).

Return Value

A reference to the new [XLSheet](#) object.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

[Overload List](#)

Clear Method

Removes all items from the collection.

Syntax

Visual Basic (Declaration)

```
Public Sub Clear()
```

C#

```
public void Clear()
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

Contains Method

Determines whether the collection contains a specific [XLSheet](#) object.

Overload List

Overload	Description
Contains(XLSheet)	Determines whether the collection contains a specific XLSheet object.
Contains(String)	Determines whether the collection contains an XLSheet object with a given name.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

Contains(XLSheet) Method

The object to look for in the collection.

Determines whether the collection contains a specific [XLSheet](#) object.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function Contains(_ ByVal sheet As XLSheet _) As System.Boolean</pre>	
C#	
<pre>public System.bool Contains(XLSheet sheet)</pre>	

Parameters

sheet

The object to look for in the collection.

Return Value

True if the collection contains the object; False otherwise.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

[Overload List](#)

Contains(String) Method

The name of the sheet to look for (case-insensitive).

Determines whether the collection contains an [XLSheet](#) object with a given name.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Contains( _  
    ByVal sheetName As System.String _  
) As System.Boolean
```

C#

```
public System.bool Contains(  
    System.string sheetName  
)
```

Parameters

sheetName

The name of the sheet to look for (case-insensitive).

Return Value

True if the collection contains an [XLSheet](#) with the given [sheetName](#); False otherwise.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

[Overload List](#)

IndexOf Method

Gets the index of a given [XLSheet](#) in the collection.

Overload List

Overload	Description
IndexOf(XLSheet)	Gets the index of a given XLSheet in the collection.
IndexOf(String)	Gets the position of the sheet with the specified name in the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

IndexOf(XLSheet) Method

The object to look for.

Gets the index of a given [XLSheet](#) in the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function IndexOf(_ ByVal sheet As XLSheet _) As System.Integer</pre>	
C#	
<pre>public System.int IndexOf(XLSheet sheet)</pre>	

Parameters

sheet

The object to look for.

Return Value

The index of the object in the collection, or -1 if the object is not a member of the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)
[XLSheetCollection Members](#)
[Overload List](#)

IndexOf(String) Method

Name of the object to look for (case-insensitive).

Gets the position of the sheet with the specified name in the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function IndexOf(_ ByVal name As System.String _) As System.Integer</pre>	
C#	
<pre>public System.int IndexOf(System.string name)</pre>	

Parameters

name

Name of the object to look for (case-insensitive).

Return Value

The index of the sheet in the collection, or -1 if the sheet can't be found in the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)
[XLSheetCollection Members](#)
[Overload List](#)

Insert Method

Creates a new [XLSheet](#) and inserts it at a specific position in the collection.

Overload List

Overload	Description
Insert(Int32)	Creates a new XLSheet and inserts it at a specific position in the collection.
Insert(Int32,XLSheet)	Inserts an XLSheet object into the collection at the specified position.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)
[XLSheetCollection Members](#)

Insert(Int32) Method

Index where the new [XLSheet](#) will be added.

Creates a new [XLSheet](#) and inserts it at a specific position in the collection.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Insert( _  
    ByVal index As System.Integer _  
) As XLSheet
```

C#

```
public XLSheet Insert(  
    System.int index  
)
```

Parameters

index

Index where the new [XLSheet](#) will be added.

Return Value

A reference to the new [XLSheet](#) object.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)
[XLSheetCollection Members](#)
[Overload List](#)

Insert(Int32,XLSheet) Method

Position where the item will be inserted.

[XLSheet](#) object to add to the collection.

Inserts an [XLSheet](#) object into the collection at the specified position.

Syntax

Visual Basic (Declaration)

```
Public Overloads Function Insert( _  
    ByVal index As System.Integer, _  
    ByVal sheet As XLSheet _  
) As XLSheet
```

C#

```
public XLSheet Insert(  
    System.int index,  
    XLSheet sheet  
)
```

Parameters

index

Position where the item will be inserted.

sheet

[XLSheet](#) object to add to the collection.

Return Value

A reference to the object that was added to the collection (in this case, always the [sheet](#) parameter).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

[Overload List](#)

Remove Method

Removes an [XLSheet](#) from the collection.

Overload List

Overload	Description
Remove(XLSheet)	Removes an XLSheet from the collection.
Remove(String)	Removes the XLSheet with the specified name from the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

Remove(XLSheet) Method

The [XLSheet](#) object to remove from the collection.

Removes an [XLSheet](#) from the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Sub Remove(_ ByVal sheet As XLSheet _)</pre>	
C#	
<pre>public void Remove(XLSheet sheet)</pre>	

Parameters

sheet

The [XLSheet](#) object to remove from the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

[Overload List](#)

Remove(String) Method

Name of the sheet to remove from the collection (case-insensitive).

Removes the [XLSheet](#) with the specified name from the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Sub Remove(_ ByVal name As System.String _)</pre>	
C#	
<pre>public void Remove(System.string name)</pre>	

Parameters

name

Name of the sheet to remove from the collection (case-insensitive).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)
[XLSheetCollection Members](#)
[Overload List](#)

RemoveAt Method

Index of the object to remove from the collection.

Removes the [XLSheet](#) at a specific index from the collection.

Syntax

Visual Basic (Declaration)

```
Public Sub RemoveAt( _  
    ByVal index As System.Integer _  
)
```

C#

```
public void RemoveAt(  
    System.int index  
)
```

Parameters

index

Index of the object to remove from the collection.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference






[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

Properties

For a list of all members of this type, see [XLSheetCollection members](#).

Public Properties

	Name	Description
	Book	Gets a reference to the C1XLBook that owns the collection.
	Count	Gets the number of XLSheet objects in the collection.
	FirstIndex	Gets or sets the index of the sheet that is initially opened when a C1XLBook file is loaded into Excel.
	Item	Overloaded. Gets the XLSheet at a given position in the collection.
	SelectedIndex	Gets or sets the index of the sheet that is selected when a C1XLBook file is loaded into Excel.

[Top](#)

See Also

Reference

[XLSheetCollection Class](#)

[C1.C1Excel Namespace](#)

Book Property

Gets a reference to the [C1XLBook](#) that owns the collection.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Book As C1XLBook</code>	
C#	
<code>public C1XLBook Book {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

Count Property

Gets the number of [XLSheet](#) objects in the collection.

Syntax

Visual Basic (Declaration)	
<code>Public ReadOnly Property Count As System.Integer</code>	
C#	
<code>public System.int Count {get;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

FirstIndex Property

Gets or sets the index of the sheet that is initially opened when a [C1XLBook](#) file is loaded into Excel.

Syntax

Visual Basic (Declaration)

```
Public Property FirstIndex As System.Integer
```

C#

```
public System.int FirstIndex {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

Item Property

Gets the [XLSheet](#) at a given position in the collection.

Overload List

Overload	Description
Item(Int32)	Gets the XLSheet at a given position in the collection.
Item(String)	Gets the XLSheet with the given name (case-insensitive).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

Item(Int32) Property

Gets the [XLSheet](#) at a given position in the collection.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads ReadOnly Property Item(_ ByVal index As System.Integer _) As XLSheet</pre>	
C#	
<pre>public XLSheet Item(System.int index) {get;}</pre>	

Parameters

index

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

[Overload List](#)

Item(String) Property

Gets the [XLSheet](#) with the given name (case-insensitive).

Syntax

Visual Basic (Declaration)

```
Public Overloads ReadOnly Property Item( _  
    ByVal name As System.String _  
) As XLSheet
```

C#

```
public XLSheet Item(  
    System.string name  
) {get;}
```

Parameters

name

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

[Overload List](#)

SelectedIndex Property

Gets or sets the index of the sheet that is selected when a [C1XLBook](#) file is loaded into Excel.

Syntax

Visual Basic (Declaration)

```
Public Property SelectedIndex As System.Integer
```

C#

```
public System.Int SelectedIndex {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLSheetCollection Class](#)

[XLSheetCollection Members](#)

XLStyle

Contains style elements used to define the appearance of the cells.

Object Model

XLStyle

Syntax

Visual Basic (Declaration)

```
Public Class XLStyle
```

C#

```
public class XLStyle
```

Remarks

Each **XLStyle** object may define one or more of the following elements: font, format, background color, background pattern, foreground color, alignment, text direction, and word wrapping.

When displaying a cell, Excel combines the row, column, and cell styles and merges the style elements defined in each one in order to determine how the cell should be displayed. The precedence of the styles is: (1) cell, (2) row, (3) column, (4) default style.

Every [XLStyle](#) belongs to a [C1XLBook](#), and may be assigned to one or more [XLRow](#), [XLColumn](#), and [XLCell](#) objects through their [Style](#) property.

Inheritance Hierarchy

System.Object

C1.C1Excel.XLStyle

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Members](#)

[C1.C1Excel Namespace](#)

Overview

Contains style elements used to define the appearance of the cells.

Object Model

XLStyle

Syntax

Visual Basic (Declaration)

```
Public Class XLStyle
```

C#

```
public class XLStyle
```

Remarks

Each [XLStyle](#) object may define one or more of the following elements: font, format, background color, background pattern, foreground color, alignment, text direction, and word wrapping.

When displaying a cell, Excel combines the row, column, and cell styles and merges the style elements defined in each one in order to determine how the cell should be displayed. The precedence of the styles is: (1) cell, (2) row, (3) column, (4) default style.

Every [XLStyle](#) belongs to a [C1XLBook](#), and may be assigned to one or more [XLRow](#), [XLColumn](#), and [XLCell](#) objects through their [Style](#) property.

Inheritance Hierarchy

System.Object

C1.C1Excel.XLStyle

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Members](#)


[C1.C1Excel Namespace](#)

Members

[Properties](#) [Methods](#)

The following tables list the members exposed by [XLStyle](#).



















Public Constructors






	Name	Description
	XLStyle Constructor	Creates a new instance of XLStyle and adds it to the specified C1XLBook .

[Top](#)

Public Properties








	Name	Description
--	------	-------------

	AlignHorz	Gets or sets the horizontal alignment for this XLStyle .
	AlignVert	Gets or sets the vertical alignment for this XLStyle .
	BackColor	Gets or sets the background color for this XLStyle .
	BackPattern	Gets or sets the background pattern for this XLStyle .
	BorderBottom	Gets or sets the line style used to draw the bottom border.
	BorderColorBottom	Gets or sets the color used to draw the bottom border.
	BorderColorLeft	Gets or sets the color used to draw the left border.
	BorderColorRight	Gets or sets the color used to draw the right border.
	BorderColorTop	Gets or sets the color used to draw the top border.
	BorderLeft	Gets or sets the line style used to draw the left border.
	BorderRight	Gets or sets the line style used to draw the right border.
	BorderTop	Gets or sets the line style used to draw the top border.
	Diagonal	Gets or sets which diagonal lines to display (none, forward, backward).
	DiagonalColor	Gets or sets the color used to draw the diagonal lines.
	DiagonalStyle	Gets or sets the line style used to draw the diagonal lines.
	Font	Gets or sets the font for this XLStyle .
	ForeColor	Gets or sets the foreground color for this XLStyle .
	Format	Gets or sets the format associated with this XLStyle .

	Indent	Gets or sets the indent cell contents for this XLStyle .
	Locked	Gets or sets whether the cell should be locked for editing when the XLSheet is protected.
	PatternColor	Gets or sets the color of the background pattern for this XLStyle .
	Rotation	Gets or sets the rotation for this XLStyle , in degrees.
	WordWrap	Gets or sets whether this XLStyle causes cell contents to wrap.

[Top](#)

Public Methods

	Name	Description
	Clone	Creates a new XLStyle object that is a copy of the current instance.
	Equals	Determines whether two XLStyle objects are equivalent.
	FormatDotNetToXL	Overloaded. Converts a .NET-style format string into an Excel format string.
	FormatXLToDotNet	Overloaded. Converts an Excel-style format string into a .NET-style format string.
	GetHashCode	Serves as a hash function suitable for use in hashing algorithms and data structures like a hash table.
	SetBorderColor	Sets the border color for this XLStyle .
	SetBorderStyle	Sets the border style for this XLStyle .

[Top](#)

See Also

Reference

[XLStyle Class](#)

[C1.C1Excel Namespace](#)

XLStyle Constructor

Parent [C1XLBook](#).

Creates a new instance of [XLStyle](#) and adds it to the specified [C1XLBook](#).

Syntax

Visual Basic (Declaration)

```
Public Function New( _  
    ByVal book As C1XLBook _  
)
```

C#

```
public XLStyle(  
    C1XLBook book  
)
```

Parameters

book

Parent [C1XLBook](#).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

Methods

For a list of all members of this type, see [XLStyle members](#).

Public Methods

	Name	Description
≡	Clone	Creates a new XLStyle object that is a copy of the current instance.
≡	Equals	Determines whether two XLStyle objects are equivalent.
≡ S	FormatDotNetToXL	Overloaded. Converts a .NET-style format string into an Excel format string.
≡ S	FormatXLToDotNet	Overloaded. Converts an Excel-style format string into a .NET-style format string.
≡	GetHashCode	Serves as a hash function suitable for use in hashing algorithms and data structures like a hash table.
≡	SetBorderColor	Sets the border color for this XLStyle .
≡	SetBorderStyle	Sets the border style for this XLStyle .

[Top](#)

See Also

Reference

[XLStyle Class](#)

[C1.C1Excel Namespace](#)

Clone Method

Creates a new [XLStyle](#) object that is a copy of the current instance.

Syntax

Visual Basic (Declaration)	
Public Function Clone() As XLStyle	

C#

```
public XLStyle Clone()
```

Return Value

A new [XLStyle](#) object that is a copy of the current instance.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

Equals Method

[XLStyle](#) object to compare to the current instance.

Determines whether two [XLStyle](#) objects are equivalent.

Syntax

Visual Basic (Declaration)

```
Public Overrides Function Equals( _  
    ByVal obj As System.Object _  
) As System.Boolean
```

C#

```
public override System.bool Equals(  
    System.object obj  
)
```

Parameters

obj

[XLStyle](#) object to compare to the current instance.

Return Value

True if the styles are equivalent, False otherwise.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

FormatDotNetToXL Method

Converts a .NET-style format string into an Excel format string.

Overload List

Overload	Description
FormatDotNetToXL(String,Type)	Converts a .NET-style format string into an Excel format string.
FormatDotNetToXL(String,Type,CultureInfo)	Converts a .NET-style format string into an Excel format string.
FormatDotNetToXL(String)	Converts a .NET-style format string into an Excel format string.
FormatDotNetToXL(String,CultureInfo)	Converts a .NET-style format string into an Excel format string.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

FormatDotNetToXL(String,Type) Method

.NET-style format to convert.

Data type to be formatted.

Converts a .NET-style format string into an Excel format string.

Syntax

Visual Basic (Declaration)

```
Public Overloads Shared Function FormatDotNetToXL( _  
    ByVal fmt As System.String, _  
    ByVal dataType As System.Type _  
) As System.String
```

C#

```
public static System.string FormatDotNetToXL(  
    System.string fmt,  
    System.Type dataType  
)
```

Parameters

fmt

.NET-style format to convert.

dataType

Data type to be formatted.

Return Value

An Excel-style format string.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)
[XLStyle Members](#)
[Overload List](#)

FormatDotNetToXL(String,Type,CultureInfo) Method

.NET-style format to convert.

Data type to be formatted.

The **System.Globalization.CultureInfo** object.

Converts a .NET-style format string into an Excel format string.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Shared Function FormatDotNetToXL(_ ByVal fmt As System.String, _ ByVal dataType As System.Type, _ ByVal culture As System.Globalization.CultureInfo _) As System.String</pre>	
C#	
<pre>public static System.string FormatDotNetToXL(System.string fmt, System.Type dataType, System.Globalization.CultureInfo culture)</pre>	

Parameters

fmt

.NET-style format to convert.

dataType

Data type to be formatted.

culture

The **System.Globalization.CultureInfo** object.

Return Value

An Excel-style format string.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

[Overload List](#)

FormatDotNetToXL(String) Method

.NET-style format to convert.

Converts a .NET-style format string into an Excel format string.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Shared Function FormatDotNetToXL(_ ByVal fmt As System.String _) As System.String</pre>	
C#	
<pre>public static System.string FormatDotNetToXL(System.string fmt)</pre>	

Parameters

fmt

.NET-style format to convert.

Return Value

An Excel-style format string.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

[Overload List](#)

FormatDotNetToXL(String,CultureInfo) Method

.NET-style format to convert.

The **System.Globalization.CultureInfo** object.

Converts a .NET-style format string into an Excel format string.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Shared Function FormatDotNetToXL(_ ByVal fmt As System.String, _ ByVal culture As System.Globalization.CultureInfo _) As System.String</pre>	
C#	
<pre>public static System.string FormatDotNetToXL(System.string fmt, System.Globalization.CultureInfo culture)</pre>	

Parameters

fmt

.NET-style format to convert.

culture

The **System.Globalization.CultureInfo** object.

Return Value

An Excel-style format string.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

[Overload List](#)

FormatXLToDotNet Method

Converts an Excel-style format string into a .NET-style format string.

Overload List

Overload	Description
FormatXLToDotNet(String)	Converts an Excel-style format string into a .NET-style format string.
FormatXLToDotNet(String,CultureInfo)	Converts an Excel-style format string into a .NET-style format string.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

FormatXLToDotNet(String) Method

Excel-style format to convert.

Converts an Excel-style format string into a .NET-style format string.

Syntax

Visual Basic (Declaration)

```
Public Overloads Shared Function FormatXLToDotNet( _  
    ByVal fmt As System.String _  
) As System.String
```

C#

```
public static System.string FormatXLToDotNet(  
    System.string fmt  
)
```

Parameters

fmt

Excel-style format to convert.

Return Value

A .NET-style format string.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)
[XLStyle Members](#)
[Overload List](#)

FormatXLToDotNet(String,CultureInfo) Method

Excel-style format to convert.

The **System.Globalization.CultureInfo** object.

Converts an Excel-style format string into a .NET-style format string.

Syntax

Visual Basic (Declaration)

```
Public Overloads Shared Function FormatXLToDotNet( _  
    ByVal fmt As System.String, _  
    ByVal culture As System.Globalization.CultureInfo _  
) As System.String
```

C#

```
public static System.string FormatXLToDotNet(  
    System.string fmt,  
    System.Globalization.CultureInfo culture  
)
```

Parameters

fmt

Excel-style format to convert.

culture

The **System.Globalization.CultureInfo** object.

Return Value

A .NET-style format string.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)
[XLStyle Members](#)
[Overload List](#)

GetHashCode Method

Serves as a hash function suitable for use in hashing algorithms and data structures like a hash table.

Syntax

Visual Basic (Declaration)	
Public Overrides Function GetHashCode() As System.Integer	
C#	
public override System.int GetHashCode()	

Return Value

A hash code for the current [XLStyle](#).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)
[XLStyle Members](#)

SetBorderColor Method

Color used to draw the border.

Sets the border color for this [XLStyle](#).

Syntax

Visual Basic (Declaration)	
<pre>Public Sub SetBorderColor(_ ByVal color As System.Drawing.Color _)</pre>	
C#	
<pre>public void SetBorderColor(System.Drawing.Color color)</pre>	

Parameters

color

Color used to draw the border.

Remarks

This method applies the setting to all four borders.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

SetBorderStyle Method

Line style used to draw the border.

Sets the border style for this [XLStyle](#).

Syntax

Visual Basic (Declaration)	
<pre>Public Sub SetBorderStyle(_ ByVal style As XLLineStyleEnum _)</pre>	
C#	
<pre>public void SetBorderStyle(XLLineStyleEnum style)</pre>	

Parameters

style

Line style used to draw the border.

Remarks

This method applies the setting to all four borders.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference


[XLStyle Class](#)



















[XLStyle Members](#)





Properties

For a list of all members of this type, see [XLStyle members](#).

Public Properties

	Name	Description
	AlignHorz	Gets or sets the horizontal alignment for this XLStyle .

 AlignVert	Gets or sets the vertical alignment for this XLStyle .
 BackColor	Gets or sets the background color for this XLStyle .
 BackPattern	Gets or sets the background pattern for this XLStyle .
 BorderBottom	Gets or sets the line style used to draw the bottom border.
 BorderColorBottom	Gets or sets the color used to draw the bottom border.
 BorderColorLeft	Gets or sets the color used to draw the left border.
 BorderColorRight	Gets or sets the color used to draw the right border.
 BorderColorTop	Gets or sets the color used to draw the top border.
 BorderLeft	Gets or sets the line style used to draw the left border.
 BorderRight	Gets or sets the line style used to draw the right border.
 BorderTop	Gets or sets the line style used to draw the top border.
 Diagonal	Gets or sets which diagonal lines to display (none, forward, backward).
 DiagonalColor	Gets or sets the color used to draw the diagonal lines.
 DiagonalStyle	Gets or sets the line style used to draw the diagonal lines.
 Font	Gets or sets the font for this XLStyle .
 ForeColor	Gets or sets the foreground color for this XLStyle .
 Format	Gets or sets the format associated with this XLStyle .
 Indent	Gets or sets the indent cell contents for this XLStyle .

	Locked	Gets or sets whether the cell should be locked for editing when the XLSheet is protected.
	PatternColor	Gets or sets the color of the background pattern for this XLStyle .
	Rotation	Gets or sets the rotation for this XLStyle , in degrees.
	WordWrap	Gets or sets whether this XLStyle causes cell contents to wrap.

[Top](#)

See Also

Reference

[XLStyle Class](#)

[C1.C1Excel Namespace](#)

AlignHorz Property

Gets or sets the horizontal alignment for this [XLStyle](#).

Syntax

Visual Basic (Declaration)

```
Public Property AlignHorz As XLAAlignHorzEnum
```

C#

```
public XLAAlignHorzEnum AlignHorz {get; set;}
```

Remarks

Set the [AlignHorz](#) property to [XLAAlignHorzEnum.Undefined](#) to suppress this style element and use the default horizontal alignment instead ([XLAAlignHorzEnum.General](#)).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

AlignVert Property

Gets or sets the vertical alignment for this [XLStyle](#).

Syntax

Visual Basic (Declaration)	
Public Property AlignVert As XLAlignVertEnum	
C#	
public XLAlignVertEnum AlignVert { get ; set ;}	

Remarks

Set the [AlignVert](#) property to [XLAlignVertEnum.Undefined](#) to suppress this style element and use the default horizontal alignment instead ([XLAlignVertEnum.Bottom](#)).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

BackColor Property

Gets or sets the background color for this [XLStyle](#).

Syntax

Visual Basic (Declaration)	
Public Property BackColor As System.Drawing.Color	
C#	
public System.Drawing.Color BackColor { get ; set ;}	

Remarks

Set the [BackColor](#) property to **Color.Transparent** to suppress this style element and use the default background color instead (white).

All colors on the [C1XLBook](#) are mapped to a palette. This means that if you assign colors to styles, save the book, and then load it back, you probably won't get exactly the same colors used when the book was created. You will get a fairly close approximation though.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

BackColor Property

Gets or sets the background pattern for this [XLStyle](#).

Syntax

Visual Basic (Declaration)	
Public Property BackPattern As XLPatternEnum	
C#	
public XLPatternEnum BackPattern { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

BorderBottom Property

Gets or sets the line style used to draw the bottom border.

Syntax

Visual Basic (Declaration)	
Public Property BorderBottom As XLLineStyleEnum	
C#	
public XLLineStyleEnum BorderBottom { get ; set ;}	

Remarks

Use the [SetBorderStyle](#) method to set all borders at once.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

BorderColorBottom Property

Gets or sets the color used to draw the bottom border.

Syntax

Visual Basic (Declaration)	
Public Property BorderColorBottom As System.Drawing.Color	
C#	
public System.Drawing.Color BorderColorBottom { get ; set ;}	

Remarks

Use the [SetBorderColor](#) method to set all borders at once.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

BorderColorLeft Property

Gets or sets the color used to draw the left border.

Syntax

Visual Basic (Declaration)	
Public Property BorderColorLeft As System.Drawing.Color	
C#	
public System.Drawing.Color BorderColorLeft { get ; set ;}	

Remarks

Use the [SetBorderColor](#) method to set all borders at once.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

BorderColorRight Property

Gets or sets the color used to draw the right border.

Syntax

Visual Basic (Declaration)	
Public Property BorderColorRight As System.Drawing.Color	
C#	
public System.Drawing.Color BorderColorRight { get ; set ;}	

Remarks

Use the [SetBorderColor](#) method to set all borders at once.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

BorderColorTop Property

Gets or sets the color used to draw the top border.

Syntax

Visual Basic (Declaration)	
Public Property BorderColorTop As System.Drawing.Color	
C#	
public System.Drawing.Color BorderColorTop { get ; set ;}	

Remarks

Use the [SetBorderColor](#) method to set all borders at once.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

BorderLeft Property

Gets or sets the line style used to draw the left border.

Syntax

Visual Basic (Declaration)	
Public Property BorderLeft As XLLineStyleEnum	
C#	
public XLLineStyleEnum BorderLeft { get ; set ;}	

Remarks

Use the [SetBorderStyle](#) method to set all borders at once.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

BorderRight Property

Gets or sets the line style used to draw the right border.

Syntax

Visual Basic (Declaration)	
Public Property BorderRight As XLLineStyleEnum	
C#	
public XLLineStyleEnum BorderRight { get ; set ;}	

Remarks

Use the [SetBorderStyle](#) method to set all borders at once.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

BorderTop Property

Gets or sets the line style used to draw the top border.

Syntax

Visual Basic (Declaration)	
Public Property BorderTop As XLLineStyleEnum	
C#	
public XLLineStyleEnum BorderTop { get ; set ;}	

Remarks

Use the [SetBorderStyle](#) method to set all borders at once.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

Diagonal Property

Gets or sets which diagonal lines to display (none, forward, backward).

Syntax

Visual Basic (Declaration)	
Public Property Diagonal As XLDiagonalFlags	
C#	
public XLDiagonalFlags Diagonal { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

DiagonalColor Property

Gets or sets the color used to draw the diagonal lines.

Syntax

Visual Basic (Declaration)	
Public Property DiagonalColor As System.Drawing.Color	
C#	
public System.Drawing.Color DiagonalColor { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

DiagonalStyle Property

Gets or sets the line style used to draw the diagonal lines.

Syntax

Visual Basic (Declaration)	
Public Property DiagonalStyle As XLLineStyleEnum	
C#	

```
public XLLineStyleEnum DiagonalStyle {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

Font Property

Gets or sets the font for this [XLStyle](#).

Syntax

Visual Basic (Declaration)

```
Public Property Font As System.Drawing.Font
```

C#

```
public System.Drawing.Font Font {get; set;}
```

Remarks

Set the [Font](#) property to null in order to suppress this style element and use the default font instead. The default font is determined by the book's [DefaultFont](#) property.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

ForeColor Property

Gets or sets the foreground color for this [XLStyle](#).

Syntax

Visual Basic (Declaration)

```
Public Property ForeColor As System.Drawing.Color
```

C#

```
public System.Drawing.Color ForeColor {get; set;}
```

Remarks

Set the [ForeColor](#) property to **Color.Transparent** to suppress this style element and use the default foreground color instead (black).

All colors on the [C1XLBook](#) are mapped to a palette. This means that if you assign colors to styles, save the book, and then load it back, you probably won't get exactly the same colors used when the book was created. You will get a fairly close approximation though.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

Format Property

Gets or sets the format associated with this [XLStyle](#).

Syntax

Visual Basic (Declaration)

```
Public Property Format As System.String
```

C#

```
public System.string Format {get; set;}
```

Remarks

Excel uses formats similar, but not identical to .NET. Refer to the Excel documentation for details on how to create format strings.

You can use the [FormatXLToDotNet\(String\)](#) and [FormatDotNetToXL\(String,CultureInfo\)](#) methods to convert common Excel format strings to and from .NET format strings.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

Indent Property

Gets or sets the indent cell contents for this [XLStyle](#).

Syntax

Visual Basic (Declaration)

```
Public Property Indent As System.Integer
```

C#

```
public System.int Indent {get; set;}
```

Remarks

This property should be set to values between 0 and 255 (between 0 and 15 for Excel 2003 compatible mode), indent calculate as value multiplied by the step (similar TAB value in the text) is equal 2,5 points (180 twips).

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

Locked Property

Gets or sets whether the cell should be locked for editing when the [XLSheet](#) is protected.

Syntax

Visual Basic (Declaration)	
<code>Public Property Locked As System.Boolean</code>	
C#	
<code>public System.bool Locked {get; set;}</code>	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

PatternColor Property

Gets or sets the color of the background pattern for this [XLStyle](#).

Syntax

Visual Basic (Declaration)	
Public Property PatternColor As System.Drawing.Color	
C#	
public System.Drawing.Color PatternColor { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

Rotation Property

Gets or sets the rotation for this [XLStyle](#), in degrees.

Syntax

Visual Basic (Declaration)	
Public Property Rotation As System.Integer	
C#	
public System.int Rotation { get ; set ;}	

Remarks

This property should be set to values between 0 and 180, or 255, as explained below:

Zero means no rotation.

1-90 means 1 to 90 degrees counter-clockwise (90 causes text to be displayed in the vertical direction going up the cell).

91-180 means 1 to 90 degrees clockwise (180 causes text to be displayed in the vertical direction going down the cell).

255 causes letters to be stacked top to bottom and not rotated.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)

[XLStyle Members](#)

WordWrap Property

Gets or sets whether this [XLStyle](#) causes cell contents to wrap.

Syntax

Visual Basic (Declaration)

```
Public Property WordWrap As System.Boolean
```

C#

```
public System.bool WordWrap {get; set;}
```

Remarks

Cells that do not wrap will spill onto adjacent cells and will be kept on a single line even if they contain line-break characters.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLStyle Class](#)
[XLStyle Members](#)

XLTextShape

Represents a common text shape embedded in an [XLSheet](#).

Object Model

XLTextShape

Syntax

Visual Basic (Declaration)	
<pre>Public MustInherit Class XLTextShape Inherits XLShape</pre>	
C#	
<pre>public abstract class XLTextShape : XLShape</pre>	

Inheritance Hierarchy

System.Object

[C1.C1Excel.XLShape](#)

C1.C1Excel.XLTextShape

[C1.C1Excel.XLCommentShape](#)

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Members](#)

[C1.C1Excel Namespace](#)

Overview

Represents a common text shape embedded in an [XLSheet](#).

Object Model

XLTextShape

Syntax

Visual Basic (Declaration)	
<pre>Public MustInherit Class XLTextShape Inherits XLShape</pre>	
C#	
<pre>public abstract class XLTextShape : XLShape</pre>	

Inheritance Hierarchy

System.Object

[C1.C1Excel.XLShape](#)

C1.C1Excel.XLTextShape

[C1.C1Excel.XLCommentShape](#)

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Members](#)















[C1.C1Excel Namespace](#)














Members

[Properties](#) [Methods](#)

The following tables list the members exposed by [XLTextShape](#).

Public Properties


	Name	Description
	Bidirectional	Gets or sets a bidirectional count of text shape of this XLTextShape .
	BottomMargin	Gets or sets the bottom margin of this XLTextShape , in twips.
	Column	Gets the index of the column to which the shape is attached. (Inherited from C1.C1Excel.XLShape)
	DashedLineStyle	Gets or sets the dash style of the line or border around the shape. (Inherited from C1.C1Excel.XLShape)
	HorizAlign	Gets or sets a horizontal alignment of the text in this XLTextShape .
	Hyperlink	Gets or sets the hyperlink associated with the shape. (Inherited from C1.C1Excel.XLShape)
	Id	Gets the unique identifier of the shape. (Inherited from C1.C1Excel.XLShape)
	IsEmpty	Determines whether the shape is empty. (Inherited from C1.C1Excel.XLShape)
	LeftMargin	Gets or sets the left margin area of this XLTextShape , in twips.
	LineColor	Gets or sets the color of the border around the shape. (Inherited from C1.C1Excel.XLShape)
	LineStyle	Gets or sets the style of the line or border around the shape. (Inherited from C1.C1Excel.XLShape)
	LineWidth	Gets or sets the width of the border around the shape, in twips. (Inherited from C1.C1Excel.XLShape)
	Locked	Gets or sets a locked of this XLTextShape .
	Orientation	Gets or sets a orientation of this XLTextShape .

	Rectangle	Gets or sets the rectangle that contains the shape, in twips. (Inherited from C1.C1Excel.XLShape)
	RightMargin	Gets or sets the right margin of this XLTextShape , in twips.
	Rotation	Gets or sets the rotation of the shape, in degrees. (Inherited from C1.C1Excel.XLShape)
	Row	Gets the index of the row to which the shape is attached. (Inherited from C1.C1Excel.XLShape)
	Sheet	Gets a reference to the parent XLSheet object. (Inherited from C1.C1Excel.XLShape)
	Text	Gets or sets a reference to the text string contained in this XLTextShape .
	TextId	Gets or sets a text identifier of this XLTextShape .
	TextScale	Gets or sets a text scale of this XLTextShape .
	TextToFit	Gets or sets a fit to shape of the text flag of this XLTextShape .
	TopMargin	Gets or sets the top margin of this XLTextShape , in twips.
	VertAlign	Gets or sets a vertical alignment of the text in this XLTextShape .
	Workbook	Gets a reference to the parent C1XLBook object. (Inherited from C1.C1Excel.XLShape)
	Wrapped	Gets or sets a wrapped text of this XLTextShape .

[Top](#)

Public Methods

Name	Description
------	-------------

 Clone()	Creates a new XLShape object that is a copy of the current instance. (Inherited from C1.C1Excel.XLShape)
---	--

[Top](#)

See Also

Reference









[XLTextShape Class](#)















[C1.C1Excel Namespace](#)






Properties

For a list of all members of this type, see [XLTextShape members](#).

Public Properties

	Name	Description
	Bidirectional	Gets or sets a bidirectional count of text shape of this XLTextShape .
	BottomMargin	Gets or sets the bottom margin of this XLTextShape , in twips.
	Column	Gets the index of the column to which the shape is attached. (Inherited from C1.C1Excel.XLShape)
	DashedLineStyle	Gets or sets the dash style of the line or border around the shape. (Inherited from C1.C1Excel.XLShape)
	HorizAlign	Gets or sets a horizontal alignment of the text in this XLTextShape .
	Hyperlink	Gets or sets the hyperlink associated with the shape. (Inherited from C1.C1Excel.XLShape)
	Id	Gets the unique identifier of the shape. (Inherited from C1.C1Excel.XLShape)
	IsEmpty	Determines whether the shape is empty. (Inherited from C1.C1Excel.XLShape)

 LeftMargin	Gets or sets the left margin area of this XLTextShape , in twips.
 LineColor	Gets or sets the color of the border around the shape. (Inherited from C1.C1Excel.XLShape)
 LineStyle	Gets or sets the style of the line or border around the shape. (Inherited from C1.C1Excel.XLShape)
 LineWidth	Gets or sets the width of the border around the shape, in twips. (Inherited from C1.C1Excel.XLShape)
 Locked	Gets or sets a locked of this XLTextShape .
 Orientation	Gets or sets a orientation of this XLTextShape .
 Rectangle	Gets or sets the rectangle that contains the shape, in twips. (Inherited from C1.C1Excel.XLShape)
 RightMargin	Gets or sets the right margin of this XLTextShape , in twips.
 Rotation	Gets or sets the rotation of the shape, in degrees. (Inherited from C1.C1Excel.XLShape)
 Row	Gets the index of the row to which the shape is attached. (Inherited from C1.C1Excel.XLShape)
 Sheet	Gets a reference to the parent XLSheet object. (Inherited from C1.C1Excel.XLShape)
 Text	Gets or sets a reference to the text string contained in this XLTextShape .
 TextId	Gets or sets a text identifier of this XLTextShape .
 TextScale	Gets or sets a text scale of this XLTextShape .

	TextToFit	Gets or sets a fit to shape of the text flag of this XLTextShape .
	TopMargin	Gets or sets the top margin of this XLTextShape , in twips.
	VertAlign	Gets or sets a vertical alignment of the text in this XLTextShape .
	Workbook	Gets a reference to the parent C1XLBook object. (Inherited from C1.C1Excel.XLShape)
	Wrapped	Gets or sets a wrapped text of this XLTextShape .

[Top](#)

See Also

Reference

[XLTextShape Class](#)

[C1.C1Excel Namespace](#)

Bidirectional Property

Gets or sets a bidirectional count of text shape of this [XLTextShape](#).

Syntax

Visual Basic (Declaration)	
Public Property Bidirectional As System.Integer	
C#	
public System.int Bidirectional { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Class](#)

[XLTextShape Members](#)

BottomMargin Property

Gets or sets the bottom margin of this [XLTextShape](#), in twips.

Syntax

Visual Basic (Declaration)

```
Public Property BottomMargin As System.Integer
```

C#

```
public System.Int BottomMargin {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Class](#)

[XLTextShape Members](#)

HorizAlign Property

Gets or sets a horizontal alignment of the text in this [XLTextShape](#).

Syntax

Visual Basic (Declaration)

```
Public Property HorizAlign As XLHorizTextAlign
```

C#

```
public XLHorizTextAlign HorizAlign {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Class](#)

[XLTextShape Members](#)

LeftMargin Property

Gets or sets the left margin area of this [XLTextShape](#), in twips.

Syntax

Visual Basic (Declaration)	
Public Property LeftMargin As System.Integer	
C#	
public System.int LeftMargin { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Class](#)

[XLTextShape Members](#)

Locked Property

Gets or sets a locked of this [XLTextShape](#).

Syntax

Visual Basic (Declaration)	
----------------------------	--

Public Property Locked As System.Boolean
C#
public System.bool Locked {get; set;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Class](#)

[XLTextShape Members](#)

Orientation Property

Gets or sets a orientation of this [XLTextShape](#).

Syntax

Visual Basic (Declaration)
Public Property Orientation As XLTextOrientation
C#
public XLTextOrientation Orientation {get; set;}

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Class](#)

[XLTextShape Members](#)

RightMargin Property

Gets or sets the right margin of this [XLTextShape](#), in twips.

Syntax

Visual Basic (Declaration)

```
Public Property RightMargin As System.Integer
```

C#

```
public System.int RightMargin {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Class](#)

[XLTextShape Members](#)

Text Property

Gets or sets a reference to the text string contained in this [XLTextShape](#).

Syntax

Visual Basic (Declaration)

```
Public Property Text As System.String
```

C#

```
public System.string Text {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Class](#)

[XLTextShape Members](#)

TextId Property

Gets or sets a text identifier of this [XLTextShape](#).

Syntax

Visual Basic (Declaration)	
Public Property TextId As System.Integer	
C#	
public System.int TextId { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Class](#)

[XLTextShape Members](#)

TextScale Property

Gets or sets a text scale of this [XLTextShape](#).

Syntax

Visual Basic (Declaration)	
Public Property TextScale As System.Single	
C#	

```
public System.float TextScale {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Class](#)

[XLTextShape Members](#)

TextToFit Property

Gets or sets a fit to shape of the text flag of this [XLTextShape](#).

Syntax

Visual Basic (Declaration)

```
Public Property TextToFit As System.Boolean
```

C#

```
public System.bool TextToFit {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Class](#)

[XLTextShape Members](#)

TopMargin Property

Gets or sets the top margin of this [XLTextShape](#), in twips.

Syntax

Visual Basic (Declaration)	
Public Property TopMargin As System.Integer	
C#	
public System.int TopMargin { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Class](#)

[XLTextShape Members](#)

VertAlign Property

Gets or sets a vertical alignment of the text in this [XLTextShape](#).

Syntax

Visual Basic (Declaration)	
Public Property VertAlign As XLVertTextAlign	
C#	
public XLVertTextAlign VertAlign { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Class](#)

[XLTextShape Members](#)

Wrapped Property

Gets or sets a wrapped text of this [XLTextShape](#).

Syntax

Visual Basic (Declaration)	
Public Property Wrapped As System.Boolean	
C#	
public System.bool Wrapped { get ; set ;}	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[XLTextShape Class](#)

[XLTextShape Members](#)

Enumerations

CalculationMode

Specifies calculation mode for all formulas in the workbook.

Syntax

Visual Basic (Declaration)	
Public Enum CalculationMode Inherits System.Enum	
C#	
public enum CalculationMode : System.Enum	

Members

Member	Description
Auto	Automatic calculation mode for all formulas in the workbook.
AutoNoTable	Automatic no table calculation mode for all formulas in the workbook.
Manual	Manual calculation mode for all formulas in the workbook (F9 in MS Excel).

Inheritance Hierarchy

System.Object

System.ValueType

System.Enum

C1.C1Excel.CalculationMode

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

CompatibilityMode

Specifies limits to sheet size that correspond to specific versions of Microsoft Excel.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum CompatibilityMode Inherits System.Enum</pre>	
C#	

```
public enum CompatibilityMode : System.Enum
```

Members

Member	Description
Excel2003	Sheets may have up to 65,536 rows and 256 columns. Workbooks may have up to 4050 unique cell styles.
Excel2007	<p>Sheets may have up to 1,048,576 rows and 18,278 columns. Workbooks may have up to 65,536 unique cell styles.</p> <p>Workbooks that exceed the Excel2003 limits must be saved in OpenXml format instead of XLS.</p> <p>It may not be possible to open these workbooks with versions earlier than Excel 2007 (C1Excel will still open them correctly).</p>
NoLimits	<p>No size limits are enforced.</p> <p>It may not be possible to open these workbooks with any version of Excel (C1Excel will still open them correctly).</p>

Inheritance Hierarchy

System.Object

System.ValueType

System.Enum

C1.C1Excel.CompatibilityMode

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

ConsolidationFunction

Specifies consolidation function for sheet subtotals.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum ConsolidationFunction Inherits System.Enum</pre>	
C#	
<pre>public enum ConsolidationFunction : System.Enum</pre>	

Members

Member	Description
Average	Represents Average function.
Count	Represents Count function.
CountNums	Represents Count Numbers function.
Max	Represents Max function.
Min	Represents Min function.
Product	Represents Product function.
StdDev	Represents StdDev function.
StdDevp	Represents StdDevp function.
Sum	Represents Sum function.
Var	Represents Var function.

Varp	Represents Varp function.
-------------	---------------------------

Inheritance Hierarchy

System.Object
 System.ValueType
 System.Enum
 C1.C1Excel.ConsolidationFunction

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

FileFormat

Specifies the file format to use when loading or saving workbooks.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum FileFormat Inherits System.Enum</pre>	
C#	
<pre>public enum FileFormat : System.Enum</pre>	

Members

Member	Description
Biff8	Excel 97/2003 format (Binary Interchange File Format revision 8).

Csv	Simple Comma-separated values (CSV) format.
OpaqueBiff8	Excel 97/2003 format (Binary Interchange File Format revision 8) without parsing opaque table.
OpenXml	Office 2007/2010 format. (Compressed Xml format).
OpenXmlTemplate	Office 2007/2010 template format. (Compressed Xml format).

Inheritance Hierarchy

System.Object
 System.ValueType
 System.Enum
 C1.C1Excel.FileFormat

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

ImageScaling

Specifies how images are scaled within spreadsheet cells.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum ImageScaling Inherits System.Enum</pre>	
C#	
<pre>public enum ImageScaling : System.Enum</pre>	

Members

Member	Description
Clip	Images are clipped to fit within the cell.
None	Images are rendered in their original size, regardless of cell size.
Scale	Images are scaled to fill the cell while preserving their original aspect ratio.
Stretch	Images are stretched to fill the cell.

Inheritance Hierarchy

System.Object
 System.ValueType
 System.Enum
 C1.C1Excel.ImageScaling

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

XLAlignHorzEnum

Specifies how to align cell content horizontally within a cell.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum XLAlignHorzEnum Inherits System.Enum</pre>	

C#

```
public enum XlAlignHorzEnum : System.Enum
```

Members

Member	Description
Center	Align to cell center.
Fill	Fill cell, repeating content as necessary.
General	Align strings to the left, numbers to the right.
Justify	Justify content horizontally to span the whole cell width.
Left	Align to cell left.
Right	Align to cell right.
Selection	Align to selection center (may span multiple cells).
Undefined	Not specified (use default).

Inheritance Hierarchy

System.Object
 System.ValueType
 System.Enum
 C1.C1Excel.XlAlignHorzEnum

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1.C1Excel Namespace

XLAlignVertEnum

Specifies how to align cell content vertically within a cell.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum XLAlignVertEnum Inherits System.Enum</pre>	
C#	
<pre>public enum XLAlignVertEnum : System.Enum</pre>	

Members

Member	Description
Bottom	Align to cell bottom.
Center	Align to cell center.
Justify	Justify content vertically to span the whole cell height.
Top	Align to cell top.
Undefined	Not specified (use default).

Inheritance Hierarchy

System.Object
 System.ValueType
 System.Enum
 C1.C1Excel.XLAlignVertEnum

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

XLDiagonalFlags

Specifies the type of diagonal line to draw across the cells.

Syntax

Visual Basic (Declaration)	
<pre><System.FlagsAttribute()> Public Enum XLDiagonalFlags Inherits System.Enum</pre>	
C#	
<pre>[System.FlagsAttribute()] public enum XLDiagonalFlags : System.Enum</pre>	

Members

Member	Description
Backward	Backward diagonal (/).
Forward	Forward diagonal (\).
None	No diagonal.

Inheritance Hierarchy

System.Object
 System.ValueType
 System.Enum
 C1.C1Excel.XLDiagonalFlags

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

XLHorizTextAlign

Specifies horizontal text alignment.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum XLHorizTextAlign Inherits System.Enum</pre>	
C#	
<pre>public enum XLHorizTextAlign : System.Enum</pre>	

Members

Member	Description
Center	Text is centered.
Justify	Text is justified.
Left	Text is left aligned.
Right	Text is right aligned.

Inheritance Hierarchy

System.Object
System.ValueType

System.Enum

C1.C1Excel.XLHorizTextAlign

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

XLLineStyleEnum

Specifies the line style used for the cell borders.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum XLLineStyleEnum Inherits System.Enum</pre>	
C#	
<pre>public enum XLLineStyleEnum : System.Enum</pre>	

Members

Member	Description
Dashed	Dashed.
Dotted	Dotted.
Double	Double.
Hair	Hair (one pixel).

Medium	Medium.
MediumDashDotDotted	Medium dash-dot-dot.
MediumDashDotted	Medium dash-dot.
MediumDashed	Medium dashed.
None	No line.
SlantedMediumDashDotted	Slanted dash-dot.
Thick	Thick.
Thin	Thin.
ThinDashDotDotted	Thin dash-dot-dot.
ThinDashDotted	Thin dash-dot.

Inheritance Hierarchy

System.Object
 System.ValueType
 System.Enum
 C1.C1Excel.XLLLineStyleEnum

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

XLPatternEnum

Specifies the pattern used to fill the cell background.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum XLPatternEnum Inherits System.Enum</pre>	
C#	
<pre>public enum XLPatternEnum : System.Enum</pre>	

Members

Member	Description
DiagonalCrosshatch	Diagonal crosshatch pattern.
DiagonalStripe	Diagonal stripe pattern (/).
Gray06	6% dotted pattern.
Gray12	12% dotted pattern.
Gray25	25% dotted pattern.
Gray50	50% dotted pattern.
Gray75	75% dotted pattern.
HorizontalStripe	Horizontal stripe pattern.
None	No pattern (transparent).
ReverseDiagonalStripe	Reverse diagonal stripe pattern (\).

Solid	Solid background.
ThickDiagonalCrosshatch	Thick diagonal crosshatch pattern.
ThinDiagonalCrosshatch	Thin diagonal crosshatch pattern.
ThinDiagonalStripe	Thin diagonal stripe pattern (/).
ThinHorizontalCrosshatch	Thin horizontal crosshatch pattern.
ThinHorizontalStripe	Thin horizontal stripe pattern.
ThinReverseDiagonalStripe	Thin reverse diagonal stripe pattern (\).
ThinVerticalStripe	Thin vertical stripe pattern.
VerticalStripe	Vertical stripe pattern.

Inheritance Hierarchy

System.Object
 System.ValueType
 System.Enum
 C1.C1Excel.XLPatternEnum

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

XLPictureViewType

Specifies how images should be displayed (color, grayscale, or black and white).

Syntax

Visual Basic (Declaration)	
<pre>Public Enum XLPictureViewType Inherits System.Enum</pre>	
C#	
<pre>public enum XLPictureViewType : System.Enum</pre>	

Members

Member	Description
Auto	Display the image using the image's own color information.
BlackAndWhite	Display the image in black and white.
GrayScale	Display the image in grayscale.

Inheritance Hierarchy

System.Object
 System.ValueType
 System.Enum
 C1.C1Excel.XLPictureViewType

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

XLRangeType

The range type.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum XLRangeType Inherits System.Enum</pre>	
C#	
<pre>public enum XLRangeType : System.Enum</pre>	

Members

Member	Description
Default	Default range.
Formula	Formula range.
Indirect	Indirect range.
Name	Named range.
Offset	Offset range.
Text	Text range.

Inheritance Hierarchy

System.Object
 System.ValueType
 System.Enum
 C1.C1Excel.XLRangeType

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

XLReferenceMode

The reference mode.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum XLReferenceMode Inherits System.Enum</pre>	
C#	
<pre>public enum XLReferenceMode : System.Enum</pre>	

Members

Member	Description
A1	A1 style, default by spreadsheet.
R1C1	R1C1 style, default by formulas.

Inheritance Hierarchy

System.Object
 System.ValueType
 System.Enum
 C1.C1Excel.XLReferenceMode

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

XLReferenceType

The reference type.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum XLReferenceType Inherits System.Enum</pre>	
C#	
<pre>public enum XLReferenceType : System.Enum</pre>	

Members

Member	Description
Absolute	Absolute coordinates.
Relative	Relative coordinates.

Inheritance Hierarchy

System.Object
 System.ValueType
 System.Enum
 C1.C1Excel.XLReferenceType

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1.C1Excel Namespace

XLShapeDashedLineStyleEnum

Specifies the dash style of borders drawn around [XLShape](#) objects.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum XLShapeDashedLineStyleEnum Inherits System.Enum</pre>	
C#	
<pre>public enum XLShapeDashedLineStyleEnum : System.Enum</pre>	

Members

Member	Description
Dash	Dash style, analogue PS_DASH pen of system.
DashDot	Dash-dot style, analogue PS_DASHDOT pen of system.
DashDotDot	Dash-dot-dot style, analogue PS_DASHDOTDOT pen of system.
Dot	Dot style, analogue PS_DOT pen of system.
LongDash	Long dash style.
LongDashDot	Long dash-dot style.
LongDashDotDot	Long dash-dot-dot style.
ShortDash	Short dash style.
ShortDashDot	Short dash-dot style.
Solid	Solid (continuous) pen.

SquareDot	Square dot style.
------------------	-------------------

Inheritance Hierarchy

System.Object
 System.ValueType
 System.Enum
 C1.C1Excel.XLShapeDashedLineStyleEnum

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

XLShapeLineStyleEnum

Specifies the style of borders drawn around [XLShape](#) objects.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum XLShapeLineStyleEnum Inherits System.Enum</pre>	
C#	
<pre>public enum XLShapeLineStyleEnum : System.Enum</pre>	

Members

Member	Description
Double	Double lines of equal width.

Simple	Single line (of width LineWidth).
ThickThin	Double lines, one thick, one thin.
ThinThick	Double lines, reverse order.
Triple	Three lines, thin, thick, thin.

Inheritance Hierarchy

System.Object

System.ValueType

System.Enum

C1.C1Excel.XLShapeLineStyleEnum

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

XLTextOrientation

Specifies text orientation.

Syntax

Visual Basic (Declaration)	
Public Enum XLTextOrientation Inherits System.Enum	
C#	
public enum XLTextOrientation : System.Enum	

Members

Member	Description
Default	Default (horizontal) orientation.
LeftRotation	Text is rotated 90 degrees counterclockwise.
RightRotation	Text is rotated 90 degrees clockwise.
TopToBottom	Vertical text (top to bottom).

Inheritance Hierarchy

System.Object
 System.ValueType
 System.Enum
 C1.C1Excel.XLTextOrientation

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

XLVertTextAlign

Specifies vertical text alignment.

Syntax

Visual Basic (Declaration)	
<pre>Public Enum XLVertTextAlign Inherits System.Enum</pre>	
C#	


```
public enum XLVertTextAlign : System.Enum
```

Members

Member	Description
Bottom	Text is bottom aligned.
Center	Text is centered vertically.
Justify	Text is justified vertically.
Top	Text is top aligned.

Inheritance Hierarchy

System.Object

System.ValueType

System.Enum

C1.C1Excel.XLVertTextAlign

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

[C1.C1Excel Namespace](#)

[C1.Win.Localization Namespace](#)

[Overview](#)

[Inheritance Hierarchy](#)

See Also

Reference

[C1.C1Excel.4 Assembly](#)

